Sottosistema video
Ampio schermo da 12 pollici • Caratteri di ottima definizione - matrice di punti 7x9 • 24 linee per 80 caratteri • Visualizzazione in verde su fondo nero • Intensità variabile dello schermo • Inclinazione dello schermo di 10° per una lettura più comoda • Ampio controllo dello schermo: scrolling, lampeggiamento, blank, inversione

Sottosistema processore
La CPU basata sulla scheda BLC/SBC 80/204 • Floppy disc controller/formatter • CRT Controller • Memoria utente da 64K byte • Quattro slot per future espansione

Tasti funzione del sistema
8 tasti per il controllo dell'esecuzione del programma • I tasti non usati sono programmabili da parte dell'utente

Tasti di editing
5 tasti controllo cursore • 13 tasti speciali di editing

Pulsante Reset e caricamento bootstrap
Bootstrap residente con micro-diagnostico per il check completo del sistema in fase di inizializzazione, con successivo switch verso la memoria utente

Tastiera ASCII
58 tasti alfanumerici

PROM programmer (opzionale)
Programmazione delle PROM bipolari e delle EPROM 2708, 2716

Tasti di selezione programmi
10 tasti di selezione programmi • 2 tasti di selezione specializzabili dall'utente

Sottosistema stampante
Matrice di punti 5x7 • Stampa termica silenziosa • 80 caratteri per linea • 50 caratteri al secondo • Caratteri full e half size • Avanzamento della carta

Sottosistema disco
Sistema a due floppy con 256K byte per driver • Formato IBM soft sectored • Espandibile a quattro driver (1.024 KB)

Le caratteristiche fondamentali e nuove dello STARPLEX:

- Facilità di utilizzo, basato su una operatività da tastiera a funzioni complesse, potendo con un solo tasto, porre in esecuzione richieste dell'operatore, che prima richiedevano una più lunga procedura.
- Software potente, costituito da Fortran IV, Basic, sistema operativo, Macro Assembler, Editor, Debugger.
- Supporto completo alla linea di prodotti NS della serie BLC 8010 e dei prodotti 8080 A, 8070, 8048...
- Modularità, sia nella configurazione fisica, che nel software.
- Affidabilità elevata; viene fornita una garanzia di 90 giorni.
- Basso costo, in quanto le ampie prestazioni dello STARPLEX costano molto meno delle ormai obsolete prestazioni di analoghi sistemi della precedente generazione di sistemi di sviluppo.
- Schede di espansione BLC 80, schede utente.
- Completa integrazione Hardware-Software.
- Opzione: ISE (In System Emulator), PROM programmer.

National Semiconductor

Per un sistema globale

- **National Semiconductor**, Milano (02) 3452046/7/8/9

AGENTE PER L'ITALIA: **Repco srl**, Milano (02) 4985274-4985932-4985494, Roma (06) 8107788
DISTRIBUTORI AUTORIZZATI PER SISTEMI E SCHEDE:
- **Adelsy spa**, Milano (02) 4524651, Genova (010) 589674, Udine (0432) 26996, Padova (049) 45600, Torino (011) 539141, Roma (06) 5915417, Osimo Scalo (AN) (071) 79307
- **E.D.L. spa**, Napoli (081) 632335, Bari (080) 365461
- **Intelco**, Bologna (051) 726186, Firenze (055) 608107
- **Inter-Rep spa**, Torino (011) 752075

Per favore speditemi ulteriori informazioni su STARPLEX
National Semiconductor
Via Solferino, 19 - 20121 Milano

Nome: ____________________

Posizione: ____________________

Società: ____________________

Via: ____________ Città: ____________

SOMMARIO

Anno 3 - N° 6 - Febbraio 1980

In copertina:

Il computer nelle applicazioni gestionali
Grafica di Renato NISI.

INDICE INSERZIONISTI

DIRETTORE RESPONSABILE
Giampietro Zanga

COORDINATORE TECNICO
Marcello Montedoro

CAPO REDATTORE
Dino Bortolossi

SEGRETARIA DI REDAZIONE
Cecilia De Serio

GRAFICA E IMPAGINAZIONE
Job Line srl

COORDINAMENTO PERSONAL COMPUTER
Gloriano Rossi

DIFFUSIONE E ABBONAMENTI
Gabriella Napoli, Silvia Decari
Marco Benedetti

DIREZIONE, REDAZIONE
P.le Massari, 22 - 20125 Milano
Telefoni 68.03.68 - 68.00.54

AMMINISTRAZIONE
Via Vincenzo Monti, 15 - 20123 Milano

PUBBLICITA': Concessionario per l'Italia e l'Estero Reina & C. S.n.c.
Via Ricasoli, 2 - 20121 Milano
Tel. (02) 803101-866192 - 8050977
864.066 - Telex: 320419 BRUS I

FOTOCOMPOSIZIONE: New Comp
Via S. Michele al Carso, 5
Nova Milanese

STAMPA: Litografia del Sole srl
Buccinasco

Concessionario esclusivo per la diffusione in Italia e all'Estero:
SODIP - Via Zuretti, 25 - 20125 Milano

Spedizione in abbonamento Postale Gruppo IV/70

Prezzo della rivista L. 2.000

Numero arretrato L. 3.000
Abbonamento annuo L. 8.000
per l'Estero L. 12.000

I versamenti vanno indirizzati a: Jackson Italiana Editrice S.r.l.
Via V. Monti, 15 - 20123 Milano
mediante emissione di assegno bancario, cartolina vaglia o utilizzando il c/c Postale numero 11666203
Per i cambi d'indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500, anche in francobolli.

EDP
USA
80

ADESSO CHE COMPUTERIA E' A MILANO, VI BASTA POCO PER SAPERE COSA PUO' DARVI UNA VERA COMPUTERIA.

Computeria ha aperto a Milano un nuovo negozio: in via Moscova 24, angolo Corso di Porta Nuova.

Qui trovate l'elaboratore per tutte le esigenze e alla portata di tutti.

L'elaboratore per la gestione delle piccole aziende industriali, commerciali, di servizi.

L'elaboratore per chi ha uno studio professionale: ingegneri, commercialisti, architetti, analisti, ricercatori.

E, naturalmente, l'elaboratore per chi ama giocare con l'elaboratore, compreso il traduttore che vi traduce all'istante il vostro italiano in tutte le lingue.

Alla Computeria sono rappresentati i piú importanti costruttori del settore. Potete cosí finalmente rendervi conto della differenza tra una casa e l'altra, tra uno strumento e l'altro.

E scegliere quello che va bene per voi, con l'aiuto dei nostri tecnici.

Ma la Computeria non vi offre soltanto degli strumenti e la consulenza per aiutarvi nel loro acquisto. Vi offre un esperto servizio di assistenza tecnica sull'hardware e sul software applicativo, se e quando vi serve.

E vi dà tutto quello che vi occorre per lavorare col calcolatore: programmi applicativi, supporti magnetici.

Uno dei prossimi giorni, fate un salto alla Computeria: cosí, per curiosare.

Arrivederci presto.

COMPUTERIA®

Il Centro del Personal Computer

è marchio registrato della Unicomp S.r.l.

Computeria: 20121 Milano-Via della Moscova, 24 - Tel. 02/666503
Unicomp: 20092 Cinisello Balsamo (Milano)
palazzo Testi - Via Cantù, 20 - Tel. 02/6121041

Anni 80

Ritengo non convenzionale parlare del prossimo decennio come fondamentale per l'informatica.

Gli anni 70 sono stati caratterizzati dalla acquisizione e dal consolidamento della tecnologia LSI (Large Scale Integration) ovvero, in parole povere, dalla miniaturizzazione dei componenti elettronici col risultato che la capacità di calcolo di un sistema che una ventina di anni fa occupava una stanza, oggi è posseduta da chips della dimensione di un millimetro di lato.

Nonostante le attuali densità di impaccamento, la tendenza a costruire dispositivi a semiconduttori con geometrie più piccole non ha raggiunto ancora il proprio limite; già l'industria di semiconduttori si è posta come obiettivo degli anno 80 l'implementazione di una nuova generazione tecnologica: la VLSI ovverso Very Large Scale Integration. L'impatto di questa tecnologia sarà fondamentale da un lato per l'industria dei calcolatori in quanto sarà possibile realizzare memorie più complesse e più veloci, e dall'altro per l'industria elettronica in generale poiché saranno implementati su un singolo componente "sistemi programmabili" per applicazioni sistemistiche specifiche.

Per quanto concerne le unità di calcolo vere e proprie, già oggi sono presenti microprocessori con capacità di eleaborazione eccezionali. Entro un paio d'anni i costruttori di sistemi offriranno prodotti costruiti attorno alle nuove generazioni di microprocessori a 16 bits mentre già si stanno sudiando strutture a 32 bits. Non ci sarà a questo punto nessuna differenza tra microcomputers e minicomputers e la distinzione, se ci sarà, sarà dovuta unicamente al fatto storico.

Sul versante memorie gli sviluppi saranno indirizzati verso due obiettivi obbligati dato che, paraddossalmente, da essi dipendono l'utilizzo esteso di unità di calcolo potenti e di basso costo: memorie più veloci e basso costo dell'unità di memorizzazione.

È essenziale inoltre l'evoluzione delle memorie di massa. Anche in questo settore ci sono attualmente le premesse, rappresentate da una parte dalle bolle magnetiche e dall'altra dalla tendenza a rendere la tecnologia dei dischi competitiva nell'ambito delle applicazioni con floppy-disk (a questo proposito già vari costruttori hanno annunciato unità a dischi di dimensioni meccaniche uguali a quelle di un floppy ma con capacità di memorizzazione dell'ordine delle decina di Mbytes). Senz'altro molto ci si aspetta dalle memorie a bolle. I vantaggi sono noti: con le memorie a bolle si abbinano alcune peculiarità delle memorie a semiconduttori (basso consumo, elevato impaccamento) con quelli delle memorie magnetiche (persistenza delle informazioni a macchina spenta) ma con tempi di accesso che potranno essere portati a valori superiori a quelli delle unità a disco.

Fondamentale comunque è il fatto che le memorie a bolle sono dispositivi elettronici e che i relativi processi di fabbricazione sono abbastanza semplici mentre le unità di massa attuali sono dispositivi elettromeccanici con una funzionalità che vede parti meccaniche in movimento; per le memorie a bolle quindi si può parlare di maggiore affidabilità ma soprattutto della tendenza a far crollare i costi dei supporti magnetici come è sempre avvenuto per i dispositivi elettronici (benchè oggi non siano ancora competitive).

Nella sostanza, dunque, gli anni 80 dovrebbero realizzare lo "sfruttamento" intensivo di tecnologie consolidate.

Eppure nonostante questo gli anni 80 saranno rivoluzionari per l'informatica: sistemi sempre più piccoli, meno costosi e tuttavia sempre più efficienti appariranno sul mercato.

Se questo da un lato ottimizzerà il rapporto prodotto-utente in quanto è ragionevole pensare che a fronte di una diversificazione di prodotti sarà possibile individuare la combinazione più opportuna per le proprie esigenze nonché per le proprie possibilità (sia a livello aziendale che a livello individuale), dall'altro lato, l'utilizzo di stumenti informatici da parte di settori estramamente vasti e compositi, significherà il passaggio da una informatica elitaria ed accentrata ad una informatica di massa e distribuita.

Il riconoscimento di questa trasformazione significa il riconoscimento di una potenzialità di servizi informatici che indipendentemente da quando sarà realizzata, identifica nell'abbinamento di una miriade di sistemi elaborativi di basso costo con la rete telefonica e/o radiotelevisiva la naturale evoluzione del settore informatico.

Realizzazione limitate e sperimentali in questo senso sono già state indicate in questa rubrica. Anche la stampa quotidiana ha segnalato in varie occasioni realizzazioni analoghe in Francia ed in Inghilterra giunto l'intesse non propriamente da addetti ai lavori verso l'intera questione.

Le dimensioni e la natura del problema variano, comunque, al di là di un discorso puramente tecnico, investendo piuttosto competenze politiche che, per quanto concerne la situazione italiana, a tutt'oggi l'hanno caratterizzata in termini negativi. Il futuro dell'informatica in Italia passa pertanto attraverso tutta una serie di decisioni politiche, in cui il punto di riferimento è il piano finalizzato di settore. Il rimando a questo è d'obbligo; comunque per il lettore interessato ad una lettura facilmente accessibile sugli sviluppi di "informatizzazione" progressiva della società è disponibile ora in Italia il rapporto Nora-Minc nell'informatica francese (l'editore è Bompiani, il titolo "Convivere con il calcolatore").

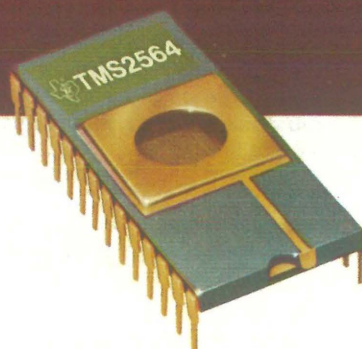
Qualunque siano le decisioni che verranno prese a livello italiano, certo è che le dimensioni ed il salto di qualità del discorso informatico impongono una maggiore partecipazione ed un maggiore sensibilità di quanti attualmente ne sono fuori in quanto la posta in gioco a lungo termine è la infomatizzazione globale della società.

TMS 2532
32K EPROM

Signal	Pin	Pin	Signal
A7	1	24	VCC
A6	2	23	A8
A5	3	22	A9
A4	4	21	VPP
A3	5	20	PD $\overline{PGM}$
A2	6	19	A10
A1	7	18	A11
A0	8	17	Q8
Q1	9	16	Q7
Q2	10	15	Q6
Q3	11	14	Q5
VSS	12	13	Q4

TMS 2564
64K EPROM

Signal	Pin	Pin	Signal
VPP	1	28	VCC
$\overline{CS}$	2	27	$\overline{CS}$
A7	3	26	VCC
A6	4	25	A8
A5	5	24	A9
A4	6	23	A12
A3	7	22	PD $\overline{PGM}$
A2	8	21	A10
A1	9	20	A11
A0	10	19	Q8
Q1	11	18	Q7
Q2	12	17	Q6
Q3	13	16	Q5
VSS	14	15	Q4

MOVING AHEAD IN MEMORIES

TMS2564

Dalla Texas Instruments la prima 64K EPROM

che si aggiunge ai dispositivi da 8K, 16K e 32K

La Texas Instruments annuncia la TMS 2564, la prima 64K EPROM sul mercato.

Questo dispositivo ha tutte le caratteristiche della famiglia delle EPROMS 5V della Texas Instruments: parola da 8 bits, funzionamento completamente statico, chip select/power down automatico e bassa dissipazione.

Compatibilità

La TMS 2564 pur essendo assemblata in un package a 28 pin è compatibile con le ROM da 64K impiccate in 24 pin e con le EPROM più piccole.

Questo dipende dal fatto che i pins dal 3 al 26 della TMS 2564 sono compatibili con i pins da 1 a 24 dei dispositivi a 24 pins. La compatibilità è raggiunta riservando sia il pin 26 che il pin 28 per alimentazione a 5V. In questa maniera, portando l'alimentazione al pin 26 si possono utilizzare sia dispositivi a 24 e 28 pin senza jumpers.

Completamente statica

Come tutte le altre EPROMS della T.I. la TMS 2564 è completamente statica: non ci sono clocks, segnali di strobes, necessità di refresh. Il tempo di accesso è uguale al temo di ciclo.

Sempre meno dissipazione di potenza

Operando con un tempo di accesso di 450 ns ma con una dissipazione di potenza di solo 840 mW cioè meno di 13 mW per il bit questo dispositivo è la EPROM con più bassa dissipazione per bit mai prodotta.

Facile da programmarsi

Il dispositivo può essere programmato con un solo impulso a livello TTL.

Inoltre può essere programmato in qualsiasi modo: individualmente, in blocchi, ramdomicamente, riducendo al minimo il tempo di programmazione.

La più ampia scelta

Con l'aggiunta della nuova TMS 2564 viene offerta al progettista una gamma di prodotti utilizzabile per ogni tipo di applicazione.

Tutte le EPROMS T.I. sono disponibili in packages da 600 mil. con pin-outs JEDEC standard.

Per maggiori informazioni contattate:
I nostri uffici commerciali di Roma, Milano, Torino.
I nostri distributori autorizzati.
Il nostro ufficio promozione commerciale, Cittaducale-Rieti.

LA FAMIGLIA DELLA EPROMS DELLA T.I.

Dispositivo	Descrizione	Alimentazione	Max Power (0°C) Operating	Standby	Tempo di accesso
TMS2564	64K	5 V	840 mW	131 mW	450 ns
TMS25L32	32K	5 V	500 mW	131 mW	450 ns
TMS2532	32K	5 V	840 mW	131 mW	450 ns
TMS2516-35	16K	5 V	525 mW	131 mW	350 ns
TMS2516	16K	5 V	525 mW	131 mW	450 ns
TMS2508-25	8K	5 V	446 mW	131 mW	250 ns
TMS2508-30	8K	5 V	446 mW	131 mW	300 ns
TMS2716	16K	+12, ±5 V	720 mW	—	450 ns
TMS27L08	8K	+12, ±5 V	580 mW	—	450 ns
TMS2708	8K	+12, ±5 V	800 mW*	—	450 ns
TMS2708-35	8K	+12, ±5 V	800 mW*	—	350 ns

*T_A = 70°C

TEXAS INSTRUMENTS
SEMICONDUTTORI ITALIA S.p.A.
Elettronica per il progresso.

NEWSLETTER

Primo censimento per mini da tavolo

Di questo censimento, curato dall'International Data Corp., fanno parte quattro categorie di calcolatori, ciascuno con proprie tipiche caratteristiche. Le categorie sono:

- Business/Professional: è il settore in più rapido sviluppo: alla fine del '78 i sistemi installati assommavano a 57.000;
- Home/Hobbyst: è il settore con la più alta intensità di unità installate (69.000 a fine '78);
- Scientific: è la categoria composta dai sistemi più costosi, dei quali la IDC ne ha conteggiati circa 27.000;
- Education: al dicembre '78 erano circa 13.000, di un costo medio intorno ai 1.000 dollari.

I dati si riferiscono alle forniture dell'industria americana. Complessivamente il parco di desktop computers ascendeva a fine '78, secondo il censimento IDC, a 226.000 unità, per un valore di 1.200 milioni di dollari. In percentuale erano queste le ripartizioni fra i costruttori:

Costruttore	Valore	Numero installazioni
Hewlett-Packard	48,4%	20,0%
IBM	15,2%	5,3%
Tektronix	10,6%	3,3%
Tandy	5,8%	31,0%
Apple	3,8%	3,0%
Wang Labs	3,5%	1,9%
Commodore	3,1%	20,8%
Processor Technology	2,0%	1,8%
Cromenco	1,5%	(+)
Pertec	1,5%	(+)
Altri	4,6%	5,0%

(+) Il dato è stato assorbito nella voce "altri".

L'impatto del TRS-80 Model II

Dopo essere diventata, come essa stessa si autodefinisce, "the biggest name in little computers", con la vendita di oltre 100 mila sistemi TRS-80 Model I nel giro di neppure due anni, la Radio Shack (divisione della Tandy Corp.) ha fatto il salto che ci si aspettava con l'introduzione di un sistema più completo, capace e sofisticato: il TRS-80 Model II.
Il TRS-80 Model II, in aggiunta ad una memoria RAM di 32 o 64 mila bytes, integra un floppy disk da 8 pollici con una capacità di mezzo milione di bytes capacità che può essere elavata a 2 megabytes con l'aggiunta di tre altri floppy-disk. Il display è di 12 pollici e visualizza 24 linee da 80 caratteri. La tastiera è di tipo professionale e può venire rimossa per facilitare il data entry. Le capacità di input/output built-in comprendono due canali RS-232C e una porta Centronics (parecchi altri boards sono in sviluppo e verranno introdotti prossimamente). I prezzi al pubblico partono da un mini-

mo di . 3450. Sulle motivazioni ed i programmi della Radio Shack nel settore del microcomputer si riporta una intervista con il presidente della stessa organizzazioine Mr. Lewis Kornfeld.

D.: Perchè avete chiamato il TRS 80 Model II "Il più importante prodotto mai fatto dalla Vostra società?

R.: In primo luogo perchè esso apre un ampio orizzonte di possibilità applicative e commerciali, tenuto conto della quantità di prodotti sofisticati che possono essere collegati. Poi perchè il nuovo sistema dimostra la capacità della Shack di operare in proprio e di stare al passo con lo stato dell'arte. Inoltre per il fatto che gli apparecchi della serie TRS-80 ci introducono in nuovi campi di marketing e a contatto con tecniche di vendita precedentemente inesplorate.

D.: Quali sono le intenzioni della Radio Shack nei confronti del TRS-80 Modello I?

R.: Noi abbiamo tutta l'intenzione di mantenerlo sul mercato e di supportarlo con periferiche e software. Esso rappresenta il "building block" dell'industria mondiale della miniinformatica personalizzata, gli utenti ne possono venire in possesso con una spesa inferiore a $ 600, e tale suo ruolo noi ci proponiamo di continuare a difendere. Non c'è senso a ritirarsi da questo segmento. Il Model I, tutto sommato, è ancora nella fase infantile, mentre il Model II ha caratteristiche tali da interessare l'utente là dove termina l'interese per il primo modello.

D.: A quali utenti si rivolge il TRS-80 Model II?

R.: Le piccole aziende commerciali ed professionisti, in qualunque campo operino, sono i più diretti interlocutori. Ma anche la grossa azienda può acquistare questo sistema, al quale potrà affidare determinati specifici compiti. Probabilmente per il Model II dovremo operare su una base più stretta di quella approcciata ed approcciabile con il Model I.

D.: Quali sono i sistemi sul mercato comparabili al TRS-80 Model II?

R.: Preferisco rispondere con un paragone: un sistema IBM 5100 configurato con un solo disco a doppia facciata e con un display di circa la metà più piccolo di quello di Model II viene venduto a 13800 dollari. Noi offriamo il TRS-80 Model II con due dischi, uno dei quali incorporato, e con una identica potenza, esattamente ad un terzo del prezzo del calcolatore IBM. Con questo non vogliamo dire che ci lanciamo alla conquista del mercato dei mini. Proprio no. I nostri metodi di marketing e di distribuzione differiscono da quelli di IBM, Hewlett-Packard ed altre case, anche se un confronto commerciale con queste marche è inevitabile.

D.: In quale posizione si pone la Radio Shack nei confronti della concorrenza riguardo alla politica dei prezzi?

R.: Il nostro principale vantaggio sulla concorrenza sta nei costi di distribuzione. In linea di massima tutti i produttori di apparecchiature d'ufficio, come i loro distributori, hanno costi di vendita estremamente alti. Per contro i sistemi TRS-80 hanno bassi costi di commercializzazione poichè vengono venduti attraverso negozi o centri EDP al dettaglio. Non abbiamo bisogno di cercare il consumatore, ma è questo che viene da noi.

In aggiunta offriamo altri vantaggi, quali: distribuzione diretta, agevolazioni nel pagamento, etc. Prevedo che la famiglia TRS-80, in modo particolare il Model II, obbligherà diverse società che producono microcomputers a rivedere le politiche dei prezzi dopo aver tentato di screditarci presso la potenziale clientela con argomentazioni basate sulla scarsa qualità dei nostri prodotti.

D.: Quale è la Sua opinione sui personal computers che utilizzano apparecchi televisivi come supporto video?

R.: Un sistema che non comprende un suo display non può essere considerato un sistema completo e sovente richiede spiacevoli compromessi. Inoltre l'utilizzazione dello schermo televisivo crea degli inconvenienti a livello di risoluzione, definizione e chiarezza che alla lunga potrebbe-

ro dar fastidio all'utente. Per questa somma di motivi nel Model II abbiamo optato per un display professionale da 12 pollici, mentre il monitor impiegato nel Model I era un diretto discendente dello schermo TV.

Notizie Apple

Si è da poco costituito in Torino un Club di appassionati di microinformatica, particolarmente interessato all'Apple II.
L'animatore dell'iniziativa è il dott. Paolo Mellana, al quale ci si può rivolgere per informazioni (Tel. 011/276222).
Il club intende organizzare riunioni e seminari su Apple per favorire lo scambio di programmi e dare informazioni, organizzare conferenze e dibattiti.

L'H-P 85 non si discosta dai tradizionali mercati dell'HP

In gennaio la Hewlett-Packard ha iniziato a vendere l'HP-85, l'esemplare di personal computer sviluppato sotto la sigla "Capricorn". L'entrata della società nel sempre più popolato mercato dei calcolatori personali avviene con un sistema di un costo indicativamente fissato in 3.950.000 e formato da: un display da 5,5 pollici, una stampante termica BUILT-IN utilizzante carta da 4 pollici, una memoria RAM da 16K, una unità a nastro integrata, una tastiera numerico-funzionale e quattro porte, di cui una per una capacità RAM addizionale da 16K e di un costo di $ 400, ed un'altra per dischi flessibili da 5,5 pollici. Il sistema si rivolge al tradizionale mercato coperto dalla Hewlett-Packard, ovvero quello per applicazioni scientifiche, ingegneristiche, finanziarie e piccolo-gestionali. L'HP-85 verrà diffusamente venduto tramite negozi, ed inizialmente sarà supportato da una dozzina di packages applicativi tra cui i seguenti: BASIC Training, General Statistics, Text Editing, Mathematics, Finance e Linear Programming.

L'interesse francese per gli ordinateurs individuels

Secondo un sondaggio compiuto dal settimanale francese "01 Informatique", fra i calcolatori personali più venduti nel vicino Paese sotto forma di kit o già montati figurano il Nascom 1 e l'AIM 65. Si rivolgono quasi esclusivamente ad amatori che abbiano una disponibilità iniziale di circa 3000 franchi. Il Nascom 1 è della inglese Nasco, che lo ha progettato a partire da un microprocessore Z80 della Zilog.
Nella sua categoria presenta un'architettura completa aperta a tutte le possibili estensioni. L'AIM 65 è invece un personal computer sviluppato dalla Rockwell a partire da un microprocessore da 8 bits, il 6501, e viene offerto già montato. Entrambi i sistemi usano il Basic e risultano dotati d'interfacce di entrata (stampante alfanumerica) e di uscita (stampante da 20 colonne per l'AIM 65 e televisore per il Nascom 1), di memorie ausiliarie e di una biblioteca di programmi orientata principalmente ai giochi.
Con un architettura sostanzialmente identica, del modello Nascom è stata recentemente messa in commercio una versione più sofisticata, la Nascom 2.
Fra i calcolatori personali venduti attraverso i grandi magazzini e i computer shop, il TRS 80 della Radio Shack ed il PET 2001 della Commodore sono quelli che nel 79 hanno riscosso i maggiori favori del pubblico francese.

Nei nuovi microprocessori della Intel priorità al software

Nonostante i proclami di uno dei suoi massimi esponenti sulla necessità di un arresto nell'incessante processo di sviluppo della microelettronica e dei microprocessori, per dare modo agli utenti di "assuefarsi " con i progressi compiuti, la intel non manifesta alcun concreto segno in tal senso. Anzi la società continua speditamente nel programma di sviluppo di nuovi prodotti. Sono due le direzioni intraprese dalla società:
- dispositivi di più elevata integrazione
- dispositivi di più elevate prestazioni

Le proposte abbonamento
1980

	Abbonamento	Prezzo
1)	Abbonamento 1980 a Elettronica Oggi	L. 29.500 anziché L. 36.000 (estero L. 41.500)
2)	Abbonamento 1980 a l'Elettronica	L. 14.000 anziché L. 15.400 (estero L. 20.000)
3)	Abbonamento 1980 a Bit (8 numeri)	L. 12.500 anziché L. 16.000 (estero L. 17.500)
4)	Abbonamento 1980 a gli Strumenti Musicali (9 numeri)	L. 13.500 anziché L. 18.000 (estero L. 19.000)
5)	Abbonamento 1980 a Elettronica Oggi + l'Elettronica	L. 42.000 anziché L. 51.400 (estero L. 59.000)
6)	Abbonamento 1980 a Elettronica Oggi + Bit	L. 40.500 anziché L. 52.000 (estero L. 57.000)
7)	Abbonamento 1980 a Elettronica Oggi + gli Strumenti Musicali	L. 41.500 anziché L. 54.000 (estero L. 58.000)
8)	Abbonamento 1980 a l'Elettronica + Bit	L. 24.500 anziché L. 31.400 (estero L. 35.000)
9)	Abbonamento 1980 a l'Elettronica + gli Strumenti Musicali	L. 25.500 anziché L. 33.400 (estero L. 36.000)
10)	Abbonamento 1980 a Bit + gli Strumenti Musicali	L. 25.000 anziché L. 34.000 (estero L.36.000)
11)	Abbonamento 1980 a Elettronica Oggi + l'Elettronica + Bit	L. 48.500 anziché L. 67.400 (estero L. 68.500)
12)	Abbonamento 1980 a l'Elettronica + Bit + gli Strumenti Musicali	L. 34.000 anziché L. 49.400 (estero L. 48.000)
13)	Abbonamento 1980 a Elettronica Oggi + Bit + gli Strumenti Musicali	L. 49.000 anziché L. 70.000 (estero L. 69.000)
14)	Abbonamento 1980 a Elettronica Oggi + l'Elettronica + gli Strumenti Musicali	L. 50.000 anziché L. 69.400 (estero L. 70.000)
15)	Abbonamento 1980 a Elettronica Oggi + l'Elettronica + Bit + gli Strumenti Musicali	L. 55.000 anziché L. 85.400 (estero L. 77.000)

Inoltre per tutto il 1980 sconto 10% su tutti i libri editi o distribuiti dalla Jackson Italiana Editrice.

In entrambi i casi i piani della Intel prevedono novità architettate intorno all'8086, il suo microprocessore da 16 bits.
Mettendosi sulla scia aperta con l'introduzione dell'8089 e con il prossimo lancio dell'8087, la Intel programma un supermicroprocessore con un superset di istruzioni. Il nuovo dispositivo avrà una maggiore capacità di indirizzamento della memoria ed un migliore sistema di protezione della memoria. Su piastra sarà forse possibile raggiungere i 20 megabytes. In aggiunta ad un aumento delle prestazioni, la Intel si preoccuperà del livello di integrazione e delle tecniche per rendere più facile l'impiego del dispositivo nelle configurazioni hardware.
Ciò significa più cure sul software. Un ordine di approssimazione tacitamente ammesso dalla Intel è il seguente: nei sistemi basati sull'8080 circa il 45% del costo di sviluppo riguarda il software; con i derivati dell'8086 tale incidenza salirà al 60-70%.

Previsioni nelle forniture di semiconduttori negli anni 80

Secondo uno studio della Frost & Sullivan, le forniture di semiconduttori da parte dell'industria americana nel periodo '78/'85 aumenteranno da 4,8 a 9,6 miliardi di dollari. In pratica raddoppieranno. L'incidenza dei semiconduttori discreti salità da 1,6 a 2,1 miliardi di dollari, con una progressione riferita all'intero periodo del 32%. Per contro i circuiti integrati evidenzieranno un incremento del 124%, grazie soprattutto al contributo dei circuiti di memoria, il cui consumo sfiorerà verso la metà del decennio i due miliardi di dollari ad avviso della Frost & Sullivan tale sviluppo va in primo luogo messo in relazione all'estensione nel ventaglio applicativo che abbraccerà, tra l'altro, personal computers, telecomunicazioni, auto, e tutte le apparecchiature di telecomunicazioni.
Le applicazioni EDP assorbiranno nel 1985 più della metà di tutte le forniture di semiconduttori. La tendenza all'integrazione verticale, da parte dei grossi gruppi industriali come Hewlett Packard, General Motors, Tekttronix, Digital Equipment, Eastman Kodak ed altri, viene interpretata dagli esperti della Frost & Sullivan come una "minaccia" per il mercato nel lungo termine. Nei riguardi delle industrie europee e giapponesi, lo studio esprime un parere sostanzialmente positivo, prevedendo per queste industrie una fase di consolidamento nel lungo termine.
Il rapporto della Frost & Sullivan conclude affermando che i produttori di piccoli calcolatori fabbricheranno direttamente gran parte dei semiconduttori di cui avranno necessità, mentre i fornitori di semiconduttori, con ancor più determinazione ed aggressività, muoveranno verso il mercato delle piastre e dei sistemi offerti come prodotti pronti per il consumo.

Coi 32 bit ci sarà un consolidamento industriale

Durante il prossimo decennio l'industria elettronica cambierà drasticamente volto. Il processo di verticalizzazione ascendente dei produttori di semiconduttori e di verticalizzazione discendente dei fabbricanti di mini non soltanto continuerà ma assumerà forme nuove. I costruttori di minicalcolatori abbandoneranno progressivamente i 16 bit per spostarsi su sistemi da 32 bit a livello di piastra. Da parte loro i fornitori di semiconduttori inizieranno a vendere, dopo la metà degli anni Ottanta, microcomputers da 32 bit sia su piastra che su chip, offrendo con la stessa capacità sistemi completi di unità di alimentazione e di unità periferiche come floppy discks. Parallelamente a questa evoluzione tecnica ci sarà una evoluzione economica di consolidamento industriale, la quale inizierà ad evidenziarsi con la prossima generazione di microprocessori. "I leaders degli anni Ottanta - si legge al riguardo in uno studio della Frost & Sullivan saranno quelli che per primi arriveranno sul mercato e che alla clientela proporranno un interessante aggio sui prezzi".

Nuovo reparto alla AMI per la fabbricazione di circuiti LSI progettati da clienti

L'importanza della fabbricazione di circuiti progettati da clienti ha portato la AMI a creare un reparto autonomo che si occuperà solamente della fabbricazione di circuiti custom. La AMI è infatti il maggior progettista e produttore di circuiti MOS/LSI custom.

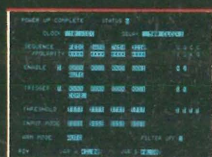
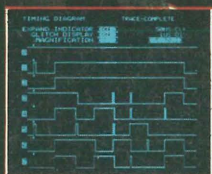
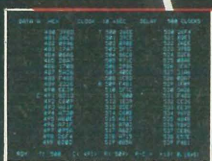
→ GOULD biomation

la più vasta gamma di logic analyzer

Il K100-D offre anche la rappresentazione dei dati Special Mode, molto utile per controllare l'Interface Bus IEEE 488.

Schermo incorporato con rappresentazione in funzione del tempo e dei dati.

Funzionamento asincrono fino a 100 MHz.

tastiera gestita da μP

1. Status Display

2. Timing Display

3. Data Display

1024 bits/can. più altrettanti per comparazioni

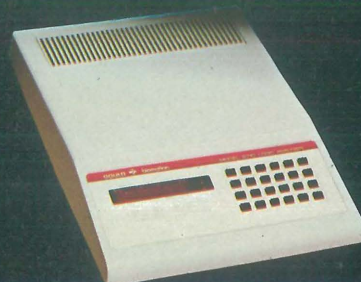
Sonde attive ad alta impedenza realizzate appositamente per il K100-D.

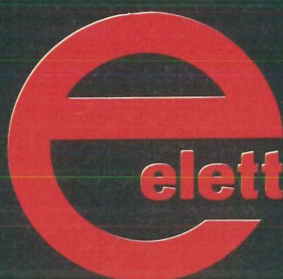
16 canali d'ingresso espandibili a 32 per risolvere anche particolari problemi connessi ai μP.

NUOVO Il Logic Analyzer K100-D è 5 volte più veloce e ha una memoria 4 volte maggiore del modello concorrente più direttamente comparabile. Non a caso questo prodotto è stato realizzato dalla Gould Biomation, la più avanzata costruttrice mondiale di tali apparecchiature. Il K100-D, con la sua possibilità di funzionamento asincrono fino a 100 MHz, i 16 canali di ingresso espandibili a 32, la memoria di 1024 bits per canale, il controllo da tastiera gestito da microprocessore, lo schermo incorporato, la rappresentazione in funzione del tempo e dei dati, il modo di funzionamento Latch selezionabile canale per canale e le sue esclusive sonde attive ad alta impedenza, si può senz'altro definire il più veloce, potente e versatile Logic Analyzer a 16 canali oggi esistente.

Modello	Freq. (MHz)	Latch (ns)	Canali	Memoria
920-D	20	10	9	256 x 9
851-D	50	5	8	512 x 8
1650-D	50	5	16	512 x 16
2710-D	10	–	27	64 x 27
8100-D	100	3	8	2048 x 8
K100-D	100	5	16 + 16	1024 x 16
9100-D	100	5	9	1024 x 9
8200-D	200	1	8	2048 x 8
168-D	10	–	1	256 x 25
DTO-1	Digital Testing Oscilloscope: 1. Logic Analyzer - 2. Storage Oscilloscope 3. Go/No-Go Comparator			

Il modello 920-D è il più economico Logic Analyzer a 20 MHz, 9 canali, 256 bits per canale, latch mode, trigger combinatoriale su tutti i canali ed è particolarmente adatto per il SERVICE dei circuiti digitali.

L 2.150.000* consegna pronta

NUOVO Il modello 2710-D è stato realizzato principalmente per l'analisi dei circuiti a microprocessori. Ha 27 canali di ingresso, 10 MHz, 64 bits per canale.

L. 1.850.000* - consegna pronta

una gamma completa di strumenti elettronici di misura

elettronucleonica s.p.a.

MILANO - Piazza De Angeli, 7 - tel. (02) 49.82.451
ROMA - Via G. Segato, 31 - tel. (06) 51.39.455

Bit

elettronucleonica S.p.A.

Desidero

- ☐ ricevere maggiori informazioni sul Logic Analyzer Gould Biomation modello ....
- ☐ avere una dimostrazione del Logic Analyzer Gould Biomation modello ....

Nome e Cognome ....

Ditta o Ente ....

Indirizzo

Sett. 79 - Pag. alla consegna, IVA esclusa - 1 \$ = 830 ± 2%

Il Vice-Presidente della AMI per i Prodotti Custom, Robert Penn, ha dichiarato : "Portando l'attività di produzione custom allo stesso livello di quella per il processo completo di progettazione e fabbricazione di circuiti custom, ci sarà possibile servire meglio quei clienti che progettano essi stessi i loro circuiti oppure affidano a terzi la progettazione.
La nostra nuova organizzazione è in grado di eseguire le operazioni di marketing, di sviluppo e di test, nonchè realizzare programmi speciali.
Con questo sistema il cliente ha maggior controllo sui costi e sui tempi della progettazione.
La progettazione interna è sempre più comune fra i clienti che utilizzano diversi nuovi circuiti custom ogni anno, che hanno progetti brevettati riservati e che possono giustificare il costo della progettazione interna.

Costituita l'associazione utenti Siemens

È stata recentemente costituita a Milano, alla presenza dei rappresentanti di 44 aziende utenti, l'Associazione Italiana Utenti Sistemi Siemens (AIUSS).
Stimolare lo scambio di idee, di informazioni e di esperienze tra gli utenti di elaboratori Siemens, promuovere, in collaborazione con la casa costruttrice, la soluzione di problemi connessi al miglior utilizzo degli impianti, suggerire lo sviluppo e la modifica di prodotti in funzione delle esigenze degli utenti, sono tra gli scopi principali della Associazione che si propone anche di stabilire dei contatti, con le associazioni utenti di sistemi Siemens già esistenti in Europa. Un altro obiettivo è la costituzione di gruppi di lavoro per i quali sono stati individuati alcuni temi di interesse generale tra i quali la trasmissione dati, il software di base, l'utilizzo e la performance dei sistemi.

CSEA: il punto sulla situazione

Si è tenuta a Torino una conferenza stampa promossa dallo CSEA (Consorzio per lo Sviluppo dell'Elettronica e l'Automazione) nella quale si sono presentati i lavori svolti nei primi sei mesi di attività e una prima definizione di una strategia imprenditoriale.
"Avviate le prime commesse - ha detto il dott. Papa direttore del Consorzio - lo sforzo maggiore è stato proteso ai primi progetti di ricerca consortili.
Sono avviati progetti di ricerca con durata pluriennale:
- sistema computerizzato per la progettazione e produzione del software realtime controllo processi che coinvolge tutte le aziende interessate CSEA
- una ricerca sulle architetture di più microelaboratori collegati tra di loro ed interagenti, utilizzando componentistica che sarà prodotta in Italia. La ricerca è fatta insieme al Politecnico di Torino, Istituto di Elettronica e al Centro Ricerche FIAT di Orbassano.

La cooperazione tra le aziende CSEA ha già visto nascere un primo prodotto, "un robot a laser per il taglio di superfici nello spazio" la cui tecnologia è all'avanguardia nel mondo".
È in atto una nuova proposta di convenzione tra il Politecnico di Torino e lo CSEA per ricerca e formazione nella meccanica, elettrotecnica, elettronica e telecomunicazioni.
Il Consorzio costituirà a breve termine una sezione CSEA per la ricerca di controlli elettronici orientali ai problemi dell'energia. Sono allo studio controlli per centrali eoliche e per l'integrazione di energie alternative e controlli per il risparmio di combustibile negli impianti di riscaldamento.
Operando in un contesto europeo lo CSEA sta assumendo contatti con aziende di altri paesi per una cooperazione tecnica e produttiva.
Lo CSEA è consapevole che sviluppo in Europa vuol anche dire decollo industriale del Mezzogiorno d'Italia contro immagini di poli sviluppati ed aree emarginate.
Pertanto sono stati già assunti contatti con imprese del Sud perchè collaborino a realizzare una unità produttiva di aziende CSEA, dalla progettazione alla produzione, nel Mezzogiorno.
Sono in corso contatti con imprese e/o consorzi di imprese operanti in altre regioni italiane per creare una rete di rapporti e collaborazioni su tutta l'area nazionale.

PERCHÈ ANCHE IL PIÙ ESIGENTE NON HA DUBBI PER SCEGLIERE GENERAL PROCESSOR?

Perchè la GP ha più esperienza. La GP è la prima azienda italiana ad aver prodotto microcomputers e personal computers; la prima in ordine cronologico e la prima per produttività. È anche la prima per la sua rapida espansione.

Perché la diffusione dei prodotti GP è conferma di qualità. I sistemi GP entrano anche "negli ambienti che contano". Esperti tecnici, istituti universitari, industrie, enti di ricerca (come il Consiglio Nazionale delle Ricerche) si affidano ogni giorno al nome GP

Perché i prodotti GP sono i più prestigiosi. Ogni progetto è fatto con in mente l'utente finale, i suoi problemi, le sue esigenze. Nessun dettaglio è trascurato e la scelta dei componenti è fatta in base a criteri estremamente rigorosi.

Perché la gamma dei prodotti GP è estremamente vasta: il nuovo Modello T è completamente espandibile in senso verticale:

- Il T/05 - con registratore audio, per l'hobby o per il calcolo scientifico
- Il T/08 dotato di minifloppy disk per la più vasta gamma di problemi applicativi
- Il T/10 - destinato alla gestione di aziende di medie dimensioni con una estesa memoria a dischetti IBM compatibili

* Il T/20 - con un grande disco da 14 o 24 Mega bytes che vi aspettereste di trovare solo su un sistema di costo molto maggiore.

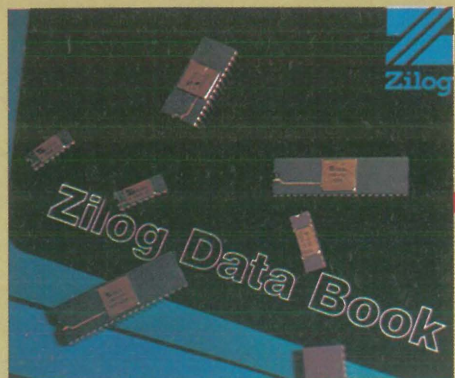
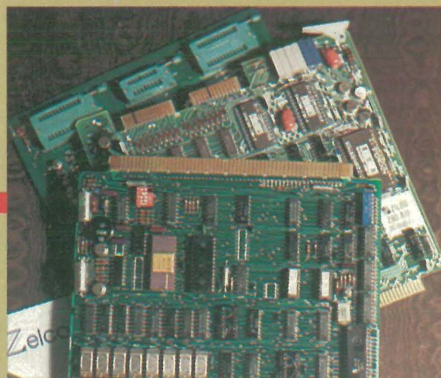
Perché i "Personal" della GP dispongono di una delle più vaste biblioteche software del mondo: Il nuovo Modello T è compatibile col famosissimo CP/M (*), il più diffuso sistema operativo a dischi oggi esistente. Sotto il CP/M (*) sono disponibili tutti i più conosciuti linguaggi di programmazione; quindi non più soltanto il BASIC, ma anche FORTRAN, COBOL, APL, PASCAL, BASEX, ASSEMBLER ecc. ecc. Il servizio software della GP è poi a vostra disposizione per personalizzare secondo le vostre necessità i numerosi programmi applicativi già realizzati o per studiarne dei nuovi. Problemi già risolti includono la contabilità generale, la gestione del magazzino, la contabilità semplificata, la prenotazione elettronica degli appuntamenti, il listino prezzi on line...

Perché l'assistenza di una ditta che opera in Italia è per forza la migliore. Una garanzia che solo una ditta italiana può offrire: la certezza di una buona e completa assistenza.

Qualunque sia il problema la risposta è una sola: General Processor. La General Processor è vicina; telefona allo 0.55 - 435527.

(*) trade mark of Digital Research, USA

E' DURO PER ALCUNI DOVER AMMETTERE DI AVER PERSO IL "QUASI MONOPOLIO". I FATTI SONO QUELLI CHE CONTANO:

LA REALTÁ DEI MICRO COMINCIA DALLA ""

DUE FISIONOMIE DIVERSE MA UNICHE SOTTO MOLTI ASPETTI

Zilog

- E' l'unica casa completamente dedicata ai micro.
- Progetta tenendo presente i problemi globali ed in funzione di questi ottimizza i particolari perciò:
- vi offre di meglio come componenti, come piastre, come sistemi e come software.
- Ha alle spalle il più forte gruppo finanziario del mondo;
- ciò le consente di avere i migliori specialisti mondiali di hardware, di software e di sistemistica e da
- la sicurezza di disporre dei finanziamenti necessari a conservare la supremazia nel proprio campo (per questo Vi sono ditte anche in italia, che già nel 1979 hanno impostato la produzione di centinaia di migliaia di macchine con lo Z80).

Zelco

- E' esclusivamente dedicata ai micro e dà un supporto che va dalla fornitura di componenti, agli accessori (terminali, stampanti etc.) per sistemi
- E' composta da specialisti e non da venditori per offrire un servizio più rapido e non una assistenza che si ferma di fronte ai problemi nuovi;
- Usa essa stessa intensamente i calcolatori che vende, li conosce a fondo e produce software sofisticato per se e per gli altri.
- Ha una organizzazione agile che le permette di non ricaricare spese eccessive sui prodotti che vende e di rispondere in "real time". (Provare per credere).
- Ha una consegna pronta; ciò significa che non dovete aspettare due mesi per un SIO o una CPU e neppure per un calcolatore.
- Vi assiste direttamente per riparazioni o consulenza qualificata.

La nostra pubblicità maggiore è fatta da tre elementi:

1 - I nostri clienti 2 - La nostra qualità 3 - I nostri prezzi

Fate i confronti: per noi basta telefonare al 02/803336/804247
(nelle zone di Roma al 06/8102836)
Zelco S.r.l. Via V. Monti 21, 20123 Milano

ZELCO TELEX 335346
PER VENDITE END-USER NELL'AREA DI SAN REMO POTETE CONTATTARE
UN.EL.CO ELETTRONICA - via Roma, 146 SAN REMO tel. 0184/883832

Considerazioni sui microcomputers per applicazioni gestionali

Parte I di **R. Williams**

Introduzione

L'utilità di un calcolatore nell'area delle piccole imprese è evidente. Per una qualsiasi azienda con un fatturato lordo annuo che vada da 250.000 dollari a 5 milioni di dollari il problema più importante non è *se* acquistare un calcolatore, ma piuttosto *quando* e *come*. Chi conduce una piccola azienda molto spesso è intimidito dall'alone di complessità e di mistero che circonda i calcolatori, a causa della non familiarità con queste nuove "macchine". Scopo di questo articolo è demistificare questa realtà, in modo da permettere una valutazione obbiettiva di quando il calcolatore è uno strumento economico in rapporto alle proprie esigenze, e di conseguenza sapere quale scelta operare.
L'argomento è trattato in una serie di quattro articoli. La Parte I tratta in modo generale l'aspetto economico connesso all'acquisizione di un calcolatore. In primo luogo si introduce un rudimentale concetto di analisi differenziale del movimento di cassa (*cash flow*), del quale poi ci si serve per descrivere gli effetti più generali della computerizzazione. Si prosegue quindi con l'analisi illustrando i vantaggi economici ed il costo di un calcolatore, tenendo presente sia il costo dell'hardware (cioè del calcolatore in sé stesso) che quello del software (cioè dei programmi).
La Parte II considera in dettaglio gli aspetti economici inerenti all'acquisizione di un calcolatore. In essa si espone più diffusamente il concetto di movimento di cassa differenziale, e quindi si analizza un'impresa ipotetica, facendo riferimento a cifre, grafici e tabelle, e fornendo in tal modo una falsariga per chi voglia applicare la medesima analisi alla propria azienda.
Le ultime due parti si occupano del problema di *come* installare un calcolatore, una volta decisi a questo passo. La Parte III, in particolare, affronta il problema della scelta, e dell'ottimizzazione, del calco-

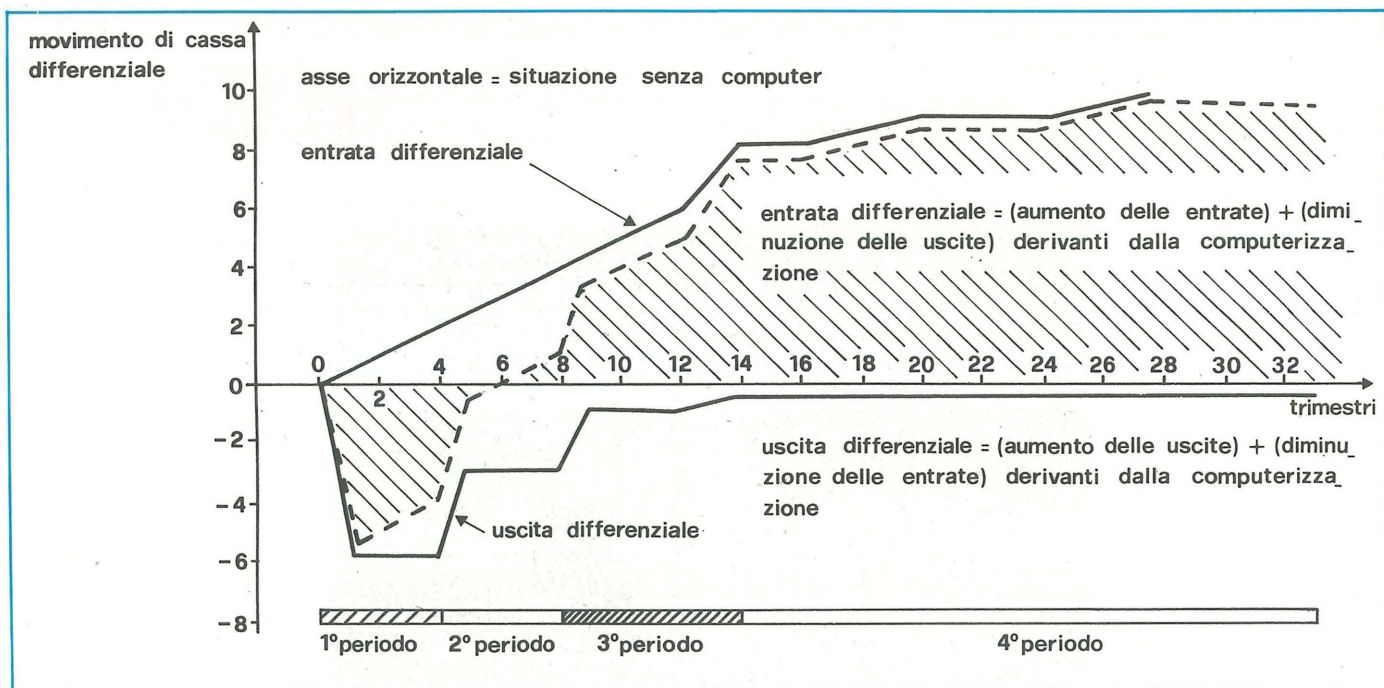



Figura 1 - Il grafico illustra l'entrata, l'uscita ed il movimento di cassa differenziali nel caso di acquisto di un calcolatore. Le scale sono ampliate per maggiore evidenza. L'asse orizzontale rappresenta la situazione di non computerizzazione. Il movimento di cassa differenziale alla fine dev'essere positivo, perché l'uscita differenziale alla fine va a zero. Il calcolatore è acquistato al tempo 0, e viene utilizzato per quantità di lavoro crescenti, mentre i costi diminuiscono, nel corso dei quattro periodi di cui si parla nel testo.

latore, dal punto di vista sia dell'hardware che del software. La Parte IV spiega che cosa avviene una volta che il calcolatore è inserito nel contesto aziendale; vi si parla quindi delle prestazioni offerte dal calcolatore, e di come gli impiegati della contabilità e i responsabili dell'amministrazione interagiscono con i programmi. Vi si suggeriscono inoltre i criteri di addestramento e tirocinio per un passaggio senza scosse dal lavoro manuale al lavoro col calcolatore.

Parte I - Considerazioni economiche generali

Il discorso generale sugli aspetti economici inerenti all'acquisizione di un calcolatore descrive in primo luogo il metodo dell'analisi differenziale del movimento di cassa, ed utilizza poi quest'analisi, in termini molto generali, per illustrare i risultati derivanti dall'acquisizione di un calcolatore, che sono validi per la grande maggioranza delle imprese. Questo discorso chiarisce inoltre che il calcolatore, senza alcun dubbio, *conviene,* e che di conseguenza la *fattibilità* è l'elemento principale nel determinare il momento in cui effettuare la computerizzazione. Si vedrà inoltre che l'acquisizione di un calcolatore è resa più fattibile nella misura in cui è individuato il giusto rapporto costo/prestazioni: cosa per fortuna non difficile, come dimostra il fatto che i costi dell'hardware e del software sono entrambi ben definiti e stabili all'interno di una oscillazione moderata del valore del dollaro.

Analisi differenziale

Il tipo di analisi utilizzato in questo articolo è l'*analisi differenziale*, nella quale gli effetti economici dovuti all'acquisizione di un calcolatore sono descritti analizzando le differenze nel movimento di cassa risultanti dall'acquisizione del calcolatore rispetto alle situazioni in cui questa non si ha. In primo luogo questo tipo di analisi richiede alcune definizioni.
Definiamo *entrata differenziale* l'aumento delle entrate, o la diminuzione delle uscite, che derivano esclusivamente dal calcolatore (ad esempio il risparmio sul costo del lavoro). Analogamente, definiamo *uscita differenziale* l'aumento delle uscite, o la diminuzione delle entrate, derivanti esclusivamente dall'acquisizione del calcolatore (ad esempio il costo del calcolatore). La differenza fra queste due quantità viene definita *movimento di cassa differenziale,* espressione che indica in breve gli effetti dell'acquisizione di un calcolatore.
Le tre definizioni sono illustrate in Figura 1.
Definiamo poi *accumulo di cassa differenziale* l'accumulo di cassa che si ha durante un determinato intervallo di tempo come diretta derivazione del movimento di cassa differenziale, o, il che è lo stesso, come diretta derivazione dell'acquisizione di un calcolatore rispetto ad una situazione in cui questa non sia stata effettuata. L'accumulo di cassa differenziale si calcola moltiplicando il movimento di cassa differenziale per l'intervallo di tempo considerato, esattamente come si calcolerebbe la quantità di acqua che (portata) si è accumulata in un secchio moltiplicando il flusso dell'acqua per l'intervallo di tempo. Se il movimento di cassa differenziale non è costante, il calcolo è leggermente più complesso, ma il concetto resta invariato. La Figura 2 illustra il risultato del calcolo, eseguito sul movimento di cassa differenziale di Figura 1.
Servendoci di queste definizioni, siamo in grado di descrivere in modo sufficientemente generalizzato gli effetti economici prevedibili dell'acquisizione di un calcolatore, indipendentemente dalle caratteristiche peculiari dell'impresa. Ad esempio, come si può vedere in Figura 1, il movimento di cassa differenzia-

le sarà in parte negativo per un primo periodo di tempo (diciamo un anno), perché durante questo periodo il calcolatore, mentre ha richiesto denaro, non svolge una quantità di lavoro proporzionata. C'è poi un secondo periodo di tempo (diciamo il secondo anno) nel quale il movimento di cassa differenziale comincia a salire rapidamente, fino a diventare positivo, via via che il calcolatore entra in fase con l'attività dell'azienda e che i costi diminuiscono. In un terzo periodo di tempo (diciamo il terzo anno) il movimento di cassa differenziale continua a salire rapidamente, dato che i costi cadono quasi a zero e il calcolatore espleta quantità sempre maggiori di lavoro. Da ultimo, in un quarto periodo di tempo (che sarà dal quarto anno in poi) il movimento di cassa differenziale continua sì a salire, ma molto più lentamente, perché la situazione si stabilizza, e i profitti aumentano lentamente con la crescita dell'impresa.

Queste stesse caratteristiche generali possono essere valutate dal punto di vista dell'accumulo di cassa differenziale, come si vede in Figura 2. Nel primo periodo si sviluppa rapidamente un passivo di cassa, che, nel secondo periodo, raggiunge la punta massima, per poi iniziare a diminuire. La diminuzione del passivo prosegue, fino ad arrivare allo zero nel secondo o nel terzo periodo (questo è il punto in cui l'investimento è ammortizzato). Da questo momento si sviluppa rapidamente un attivo, che poi continua a salire ad un ritmo che aumenta moderatamente con l'andar del tempo.

Questa analisi dell'accumulo di cassa differenziale è estremamente efficace, in quanto fa vedere che dall'acquisizione di un calcolatore *devono* risultare, in denaro vero, dei profitti.

Si noti che il movimento di cassa differenziale in definitiva diventerà positivo se il calcolatore è usato a pieno ritmo, perché i costi mensili alla fine scenderanno fin quasi a zero. Ciò significa che l'accumulo di cassa differenziale alla fine deve raggiungere il punto zero, o di ammortamento (anche se, probabilmente, dopo parecchio tempo). Quindi il calcolatore è senz'altro un investimento *a lungo termine*, nel senso che si ammortizzerà in un determinato momento. Se l'ammortamento può essere raggiunto in tre o quattro anni, possiamo considerare il calcolatore un investimento *a breve termine*.

La validità dell'analisi è ancor più sottolineata se si tiene conto delle conseguenze di un semplice ritardo nell'acquisizione del calcolatore, anche se l'investimento è considerato a lungo, e non a breve termine.

Come si può vedere in Figura 3, in questo caso si ritarderà anche l'accumulo di cassa differenziale, creando una differenza *costante* nell'accumulo di cassa, riscontrabile in *qualunque* momento dopo i primi anni. Questi costi determinati dal ritardo, che sono irreversibili, sono rappresentati in modo chiaro nella Figura 4, che mette a confronto i movimenti dell'accumulo di cassa dovuti a diversi ritardi con quello ritardo nullo. Si tenga presente che, nel lungo perio-

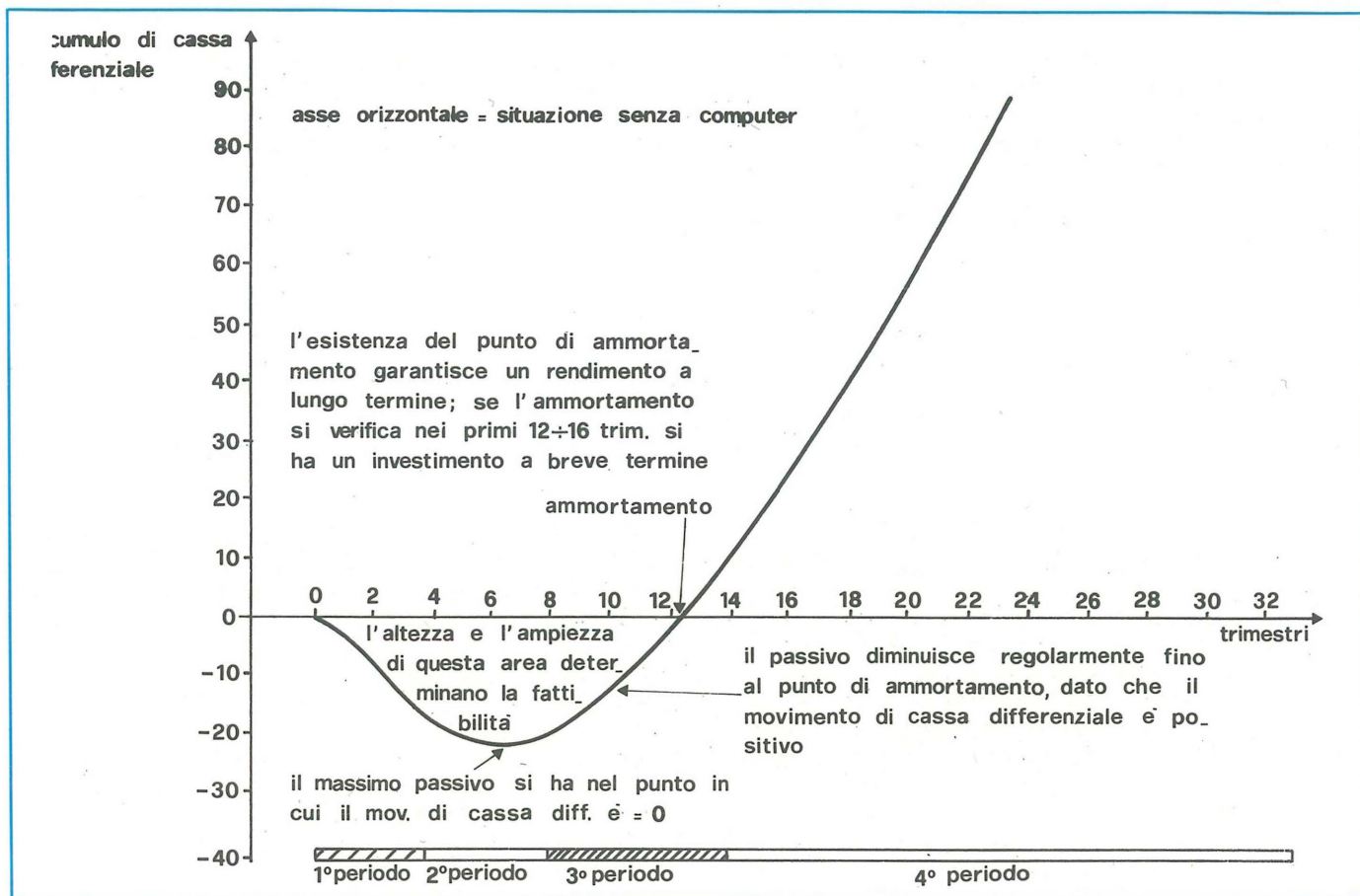



Figura 2 - Il grafico illustra l'accumulo di cassa differenziale relativo alla Figura 1. Il calcolo, che è descritto nel testo, è conosciuto in ambito tecnico come "integrale riferito al tempo" del movimento di cassa differenziale. Le scale sono arbitrarie ed ampliate per maggior chiarezza. I concetti di convenienza e fattibilità sono spiegati nel testo. Si noti l'aumento continuo dei profitti differenziali dopo il punto di ammortamento. Questi non ci sarebbero senza calcolatore.

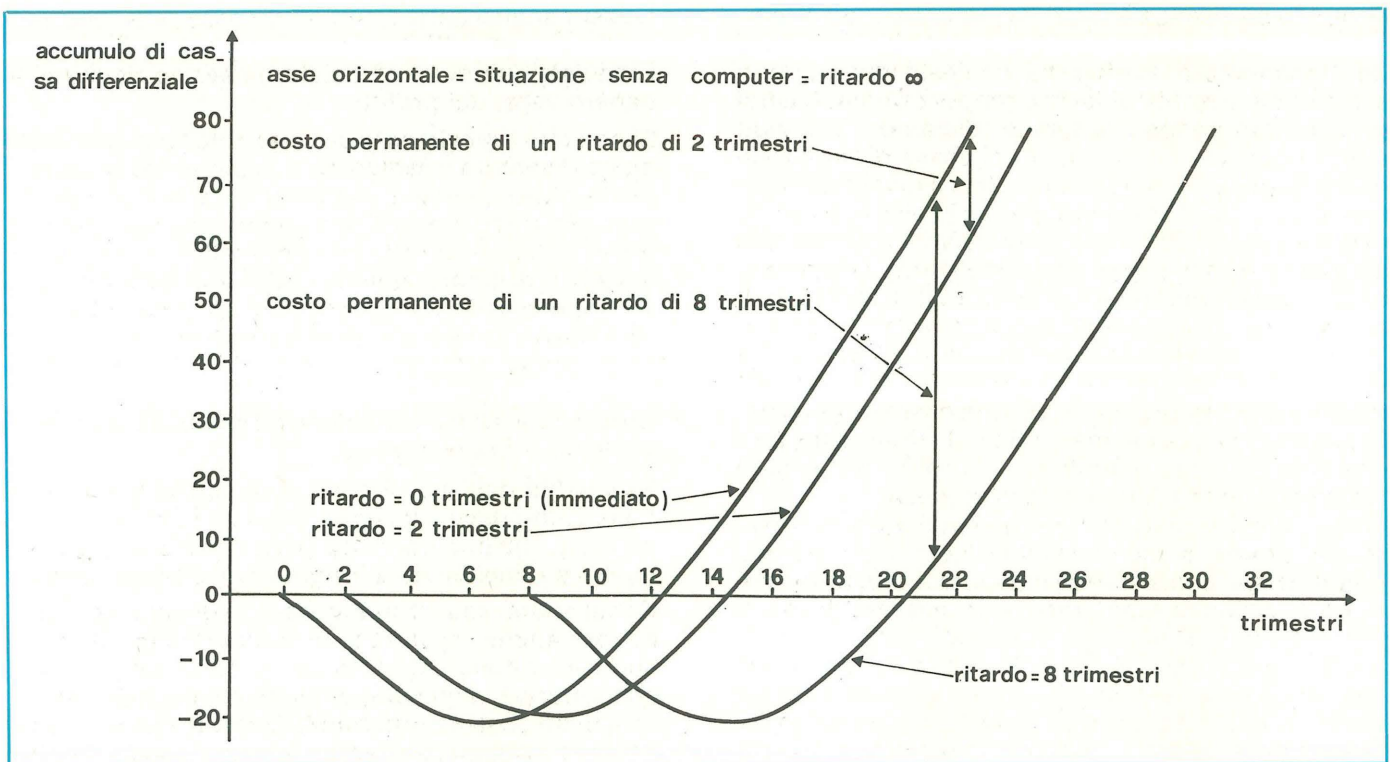



Figura 3 - Il grafico illustra gli effetti di un ritardo nell'acquisizione di un calcolatore e rappresenta l'accumulo di cassa differenziale di Figura 2 ritardato di 2 e di 8 trimestri. Gli effetti sono irreversibili e crescono regolarmente. L'asse orizzontale rappresenta la situazione di non computerizzazione (ritardo = ∞).

do, si ha un passivo nell'accumulo di cassa differenziale continuo e in costante aumento.
Si tratta di soldi, non di carta! Ed è tenendo ben presente questa pregiudiziale che la decisione di *non* acquisire *immediatamente* un calcolatore va valutata con molta attenzione, come una scelta notevolmente rischiosa, che implica grosse probabilità di perdite differenziali irreversibili e via via crescenti.
Naturalmente, esiste la possibilità che una piccola impresa non possa *permettersi* un calcolatore, quali che possano essere le conseguenze in termini di perdite differenziali, e quindi debba adattarsi alle perdite. Questa situazione si verifica quando le risorse di cassa dell'impresa non sono tali da coprire, senza uno sforzo sproporzionato, l'iniziale movimento di cassa differenziale (di andamento negativo) che risulta dai costi del calcolatore. È una situazione classica che nasce dalla sottocapitalizzazione, in cui si ha il potenziale per grossi profitti, ma non lo si può utilizzare per scarsità di capitali.
Essendo ormai chiaro che il calcolatore dà un rendimento a lungo termine, e, a volte, a breve termine, il problema più immediato è quando ci si possa permettere l'acquisto di un calcolatore. Il problema richiede un esame più particolareggiato dei vantaggi previsti (entrata differenziale), rapportati ai costi di acquisizione (uscita differenziale). Operando con l'analisi differenziale, bisogna determinare il movimento di cassa differenziale, e poi riportarlo al momento in cui le risorse dell'azienda potranno far fronte al periodo iniziale di movimento di cassa differenziale negativo. Val la pena di osservare che l'acquisizione di un calcolatore, per essere realisticamente fattibile, deve essere necessariamente redditizia a breve termine (a meno che l'impresa disponga di sufficienti capitali per supportare dei movimenti di cassa negativi sul lungo periodo!)
In altre parole, la decisione, presa razionalmente, di acquistare un calcolatore è già sufficientemente valida se è dimostrata la sola fattibilità, e nient'altro. Il problema non è se acquistare un calcolatore (a questo si è già risposto di sì), ma quando farlo; e la risposta è: "appena è possibile permetterselo".
La condizione di fattibilità più favorevole è che il passivo nell'accumulo di cassa sia il più piccolo possibile. Questo richiede che i vantaggi (entrata differenziale) siano il più possibile elevati e rapidi, e che i costi (uscita differenziale) siano il più possibile bassi e limitati nel tempo. Questi fattori sono evidentemente in contrapposizione: un basso costo mensile del calcolatore si traduce in una bassa capacità del calcolatore di produrre vantaggi economici, e viceversa. Esiste allora un equilibrio ottimale fra costo e prestazioni, e l'obbiettivo è di prevenire a questo optimum grazie ad una scelta avveduta degli elementi del calcolatore, come vedremo particolareggiatamente nella Parte III.
Comunque, dettagli a parte, specificare questo equilibrio non è difficile come ci si potrebbe aspettare, e serve ad individuare il costo di implementazione di un calcolatore in modo ben definito e circostanziato. Valutando con oculatezza hardware e software, stabiliremo un costo finale che va da 7000 a 10.000 dollari per l'hardware, un costo di 2.000 dollari per il software, ed un costo mensile di 600 dollari per la personalizzazione di programmi per un anno. Come si vedrà più avanti, queste cifre, sorprendentemente, sono in qualche misura indipendenti dalle dimensioni della piccola impresa.
Una volta stabilito il costo approssimativo dell'imple-

mentazione (uscita differenziale), bisogna valutarne i vantaggi (entrata differenziale). Pur non essendo possibile precisarli esattamente, tuttavia se ne può dare una valutazione. Il risparmio di personale riguarderà da 1/2 a 5 impiegati, che rappresentano dai 300 ai 3.000 dollari al mese. L'aumento delle vendite ed il miglioramento della gestione incrementeranno il profitto lordo dal 2% al 4%. Anche di questo punto parleremo più diffusamente in seguito.
Questi valori di uscita e di entrata differenziali, applicati ad una data azienda, servono per calcolare il movimento di cassa differenziale e l'accumulo di cassa differenziale determinati dall'acquisizione di un calcolatore. Questi a loro volta, come si è già sottolineato, decideranno della convenienza e della fattibilità del calcolatore. Nella Parte II si è usato questo procedimento per analizzare un'azienda ipotetica, allo scopo di verificare il momento in cui l'acquisto di un calcolatore è fattibile con ammortamento in 3 o 4 anni, e di spiegare che cosa si verifica a livello di cassa, e quali saranno gli eventuali costi permanenti derivanti da un ritardo.
Nel resto di questa Parte I chiariremo le cifre fornite per l'uscita differenziale prevista (costi del calcolatore) e per l'entrata differenziale prevista (vantaggi del calcolatore), facendo riferimento ad elementi relativamente ben definiti, e indipendenti dalla specifica azienda.

Costi di implementazione di un calcolatore

I costi di implementazione di un calcolatore dipendono, innanzitutto, dal fatto che l'edificio ove ha sede l'azienda possa accogliere un sistema completo, o piuttosto soltanto un terminale collegato via telefono ad un sistema remoto in time-sharing.

La soluzione time-sharing

La soluzione time-sharing, peraltro molto diffusa nel passato, sta rapidamente cadendo in disuso nelle piccole aziende. Le ragioni diventano evidenti all'esame di quattro fattori: il costo del terminale, il costo del collegamento con il calcolatore remoto, il costo della memorizzazione dei dati dell'azienda su memorie di massa, ed il costo del software.
Il costo di un terminale con CRT (console con schermo tipo TV con tastiera) e possibilità di stampa (una necessità per le piccole imprese) è circa un terzo del costo di un sistema completo. L'affitto andrà dai 125 ai 175 dollari al mese. A questo va aggiunto il costo del collegamento con il calcolatore remoto. La linea telefonica costerà circa 1 dollaro all'ora, ed il tempo di "collegamento" costerà 5 dollari all'ora nelle ore lavorative. Questi costi orari, supponendo un collegamento di 4 ore al giorno, danno un totale di 24 dollari al giorno, oppure di quasi 500 dollari al mese.
Il costo della memorizzazione su memoria di massa è legato alla quantità di dati memorizzati nel calcolatore remoto. È importante notare che i dati non possono essere trasmessi velocemente dal terminale al calcolatore e viceversa, e che il terminale CRT, di cui si è parlato in precedenza, non è dotato di memorie di massa. Pertanto il calcolatore remoto deve accumulare tutti i dati di contabilità relativi almeno ad un tri-

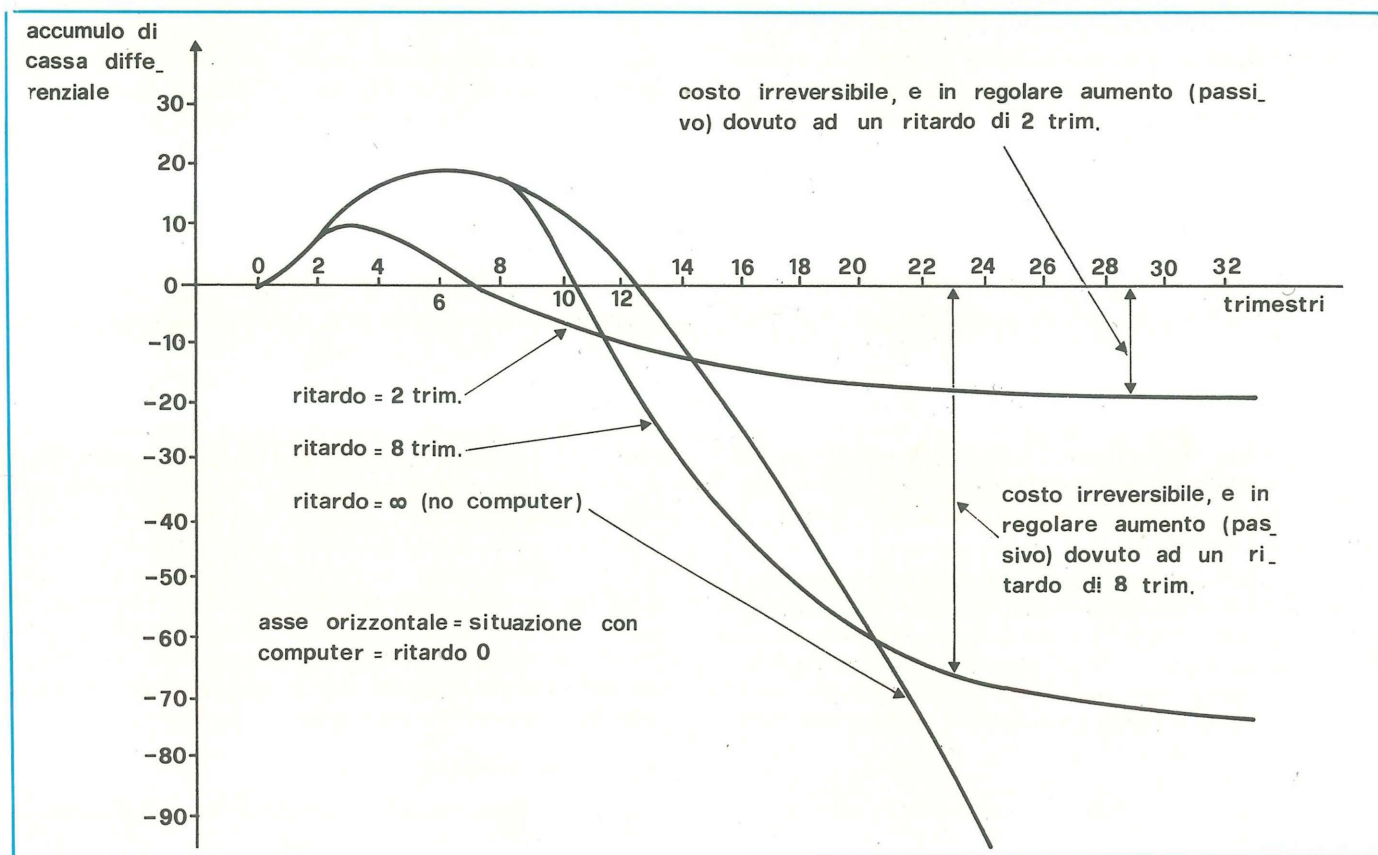



Figura 4 - Il grafico è identico a quello di Figura 3, solo che l'asse orizzontale rappresenta questa volta la situazione di computerizzazione con ritardo 0. Questa diversa prospettiva mette chiaramente in evidenza gli effetti del ritardo come passivo irreversibile e continuamente crescente dell'accumulo di cassa differenziale.

mestre, e quindi fornire un resoconto dell'archivio al termine di ogni trimestre. (Si badi che questo rende impossibile in futuro l'accesso ai medesimi dati!). La quantità di dati così accumulati può occupare facilmente l'equivalente di "due floppy disks". Una memorizzazione siffatta, di circa 500.000 "bytes", costerà all'incirca dai 150 ai 250 dollari al mese.
Bisogna poi considerare i costi del software. Benché il time-sharing offra il vantaggio di evitare il costo iniziale di certi programmi, questi programmi non possono venir modificati per usi personalizzati, non solo, ma bisogna pagare in più tutta la programmazione fatta per scopi speciali. Il risparmio iniziale sul costo del software va dai 1.000 ai 2.000 dollari: è un risparmio marginale, che con un calcolatore in sede viene compensato nel giro di pochi mesi.
Concludendo, possiamo stimare che i costi dell'hardware nella soluzione time-sharing ammontino ad 800 - 1.000 dollari al mese, calcolando un uso di quattro ore al giorno. Se anche il tempo di utilizzazione scende a mezz'ora al giorno, i costi per mese non vanno comunque al di sotto dei 400 - 500 dollari, perché in ogni caso c'è occupazione della memoria di massa del calcolatore remoto, e si ha il terminale nella sede della ditta. Se poi il tempo di utilizzo sale ad otto ore al giorno, il che potrebbe avvenire per lo sviluppo di programmi fatti dall'utente, il costo mensile arriverà fino a 1.200 - 1.400 dollari.
Come si è già detto, e ne riparleremo ancora più in dettaglio, l'hardware di un calcolatore in sede costerà 10.000 dollari al massimo, o, indicativamente, 500 dollari al mese, a seconda delle capacità finanziarie dell'azienda. Pertanto la soluzione time-sharing costerà al mese almeno quanto un calcolatore tenuto in sede, anche se poco utilizzato, e facilmente due volte tanto. Senza contare che si tratta di spese a fondo perduto, che non si convertono in beni patrimoniali, come invece accade per un calcolatore di proprietà.

La soluzione calcolatore

È chiaro a questo punto come ci si debba porre di fronte all'alternativa time-sharing/calcolatore in sede, almeno per quanto riguarda le applicazioni relative alle piccole aziende aventi accesso al normale time-sharing, e sulla base dei costi finali di un calcolatore tenuto in sede, quali sono stati in precedenza determinati.
Esaminiamo ora questi costi più in dettaglio, attraverso l'analisi dei costi generali dell'hardware, dei costi generali del software, e della prospettiva che questi costi si mantengano stabili per i prossimi anni.

Costo dell'hardware

Praticamente tutte le piccole aziende richiederanno lo stesso tipo di hardware, situazione che è resa paradossale dal fatto che anche l'insieme minimo di hardware *essenziale* è solitamente sottoutilizzato nelle imprese più piccole. (Ciò significa anche che il calcolatore si adeguerà ad esigenze via via crescenti senza ulteriori investimenti!). La configurazione minimale dell'hardware, per qualunque azienda, è costituita dalla struttura principale, cioè il calcolatore vero e proprio, più una stampante, un terminale CRT, e da dispositivi costituenti la memoria di massa.
Più precisamente, il calcolatore vero e proprio deve avere minimo 40 "kilobytes" di memoria, per poter accogliere programmi; la stampante dovrà avere introduttori frontali, per poter trattare moduli prestampati; il terminale CRT deve poter visualizzare caratteri maiuscoli e minuscoli in modo ben leggibile, ed avere una capacità di 24 linee per 80 caratteri per linea; inoltre dovrebbe avere, oltre alla tastiera standard tipo macchina da scrivere, anche una tastiera numerica. La memoria di massa dovrebbe consistere di almeno un doppio floppy disk (con una capacità di memorizzazione superiore al mezzo milione di bytes) allo scopo di poter fare la duplicazione dei dischi e di fornire una capacità di memorizzazione dimensionata su un'attività che richiede l'intervento di più di una persona.
Il costo del calcolatore vero e proprio va dai 2.000 ai 2.500 dollari, comprese le schede di memoria e le schede di ingresso/uscita. La stampante costa da 1.000 a 1.500 dollari, compreso l'introduttore frontale. Il terminale CRT costa da 1.000 a 1.500 dollari. L'unità floppy disk con doppio drive, e con una capacità di memorizzazione di mezzo milione di bytes, costa da 2.000 a 2.500 dollari, compresa l'elettronica di governo. Concludendo, il costo di un calcolatore, per il solo hardware *minimo indispensabile*, va da 6.000 ad 8.000 dollari, in media 7.000. Questi costi sono abbastanza indipendenti dalla marca e dalle varie caratteritiche particolari.
Il costo di un sistema *massimo,* è chiaro, può salire notevolmente. Comunque, per 10.000 dollari si avranno delle prestazioni che vanno oltre le ragionevoli esigenze di una piccola azienda. Già molti elementi sono sottoutilizzati: ad esempio, volendo, la stampante sarebbe in grado di stampare un libro al giorno; il terminale CRT, se usato tutto il giorno, potebbe trattare quantità enormi di dati. Il calcolatore vero e proprio non necessita di una maggior quantità di memoria o di altri dispositivi, perché è già in grado di fare calcoli in una misura da 10 a 100 volte superiore a quanto occorra in una piccola azienda. Il trattamento e la memorizzazione di dati è la funzione più importante. L'unica limitazione è data dalla capacità di memorizzazione dell'unità floppy disk, ma la si può aggirare cambiando spesso i dischi. (Il disco si può togliere o inserire nel drive come si vuole, esattamente come una cassetta di registratore).
Volendo, si può acquistare, con una maggiorazione del prezzo iniziale non superiore ai 3.000 dollari, una memoria su disco di capacità 10 volte maggiore di quella che si è vista prima. Se si ha bisogno di un sistema multiterminale, si potranno acquistare i terminali CRT in più e l'hardware ed il software necessari con una maggiorazione del prezzo iniziale non superiore a 3.000 dollari.
Concludendo, si può dire tranquillamente che un costo da 7.000 a 10.000 dollari per l'hardware coprirà le esigenze di una "piccola azienda", eccezion fatta per quelle più grandi, e che anche le esigenze di queste potranno essere soddisfatte con un operatività forse un po' noiosa, ma non molto impegnativa, come il cambio frequente dei dischi.

Costo del software

Vediamo ora i costi del software. I tipi di software necessario sono sostanzialmente due: il primo tipo, detto software operativo, ha la funzione di coordinare le attività di CRT, stampante e floppy disk con quella del calcolatore vero e proprio, e in molti casi viene fornito insieme all'hardware; il secondo tipo,

detto software applicativo, esegue di fatto il lavoro richiesto dell'utilizzatore, ad esempio la contabilità.
I programmi applicativi specifici che interessano le piccole aziende (contabilità generale, fatturazione, gestione magazzino) costeranno da 1.000 a 2.000 dollari. Anche se tale software applicativo a volte non fa esattamente quello che si vorrebbe, e sembra caro, in realtà vale effettivamente il suo prezzo. Tentare una totale programmazione in proprio sarebbe per una piccola impresa un errore che verrebbe a costare molto.

Il costo di sviluppo del software ordinario è spesso sottovalutato. Le stime fatte dalle società che sviluppano software parlano di oltre 10 dollari per linea di codice, essendo questa cifra comprensiva della documentazione, del debugging e naturalmente dell'implementazione. Quindi i programmi per applicazioni gestionali costeranno ben oltre i 10.000 dollari, e anche più.

Un costo indiretto del software è quello determinato dal ritardo dell'implementazione, in quanto si hanno degli effetti permanenti sull'accumulo di cassa differenziale, proprio come si è visto per i ritardi nell'acquisto del calcolatore. Questi ritardi possono essere molto maggiori di quanto un profano possa valutare, essendo dell'ordine di parecchi mesi.

Appare chiaro dunque che conviene comprare il maggior numero di packages applicativi di cui si ha bisogno. La programmazione in proprio andrebbe limitata a quelle applicazioni per le quali non sono disponibili dei packages standard. Inoltre tale programmazione è meglio che sia fatta gradualmente. Un programmatore che lavori a metà tempo per un anno costerà all'incirca 600 dollari al mese, e potrà svolgere da una parte il lavoro strettamente necessario, quale la personalizzazione dei programmi acquistati, dall'altra lo sviluppo completo di alcuni importanti programmi. Questo approccio dà inoltre all'azienda il tempo sufficiente per adattarsi al calcolatore, e per definire meglio gl'interventi di programmazione che occorrono; inoltre aiuta ad evitare che vengano implementate specifiche destinate ben presto a rivelarsi non più attuali, e che in potenza porterebbero ad uno spreco di denaro. Inoltre facilmente il lavoro di programmazione ha una curva di rendimento tale che i 2/3 dell'attività vengono svolti nella prima metà della giornata.

Stabilità dei costi di un calcolatore

L'ultima osservazione sui costi prende in considerazione la stabilità dei costi dell'hardware e del software durante i prossimi anni: si assume cioè che verosimilmente questi costi, fra un anno o due, siano quasi invariati rispetto a quelli attuali. L'aspettare un anno non comporterà per una piccola azienda un risparmio apprezzabile.
Si pensa a volte, basandosi sull'osservazione del grosso calo dei costi delle calcolatrici tascabili e degli hobby computers verificatosi negli ultimi anni, che lo stesso fenomeno si avrà per un completo sistema gestionale, e cioè che entro un anno si potrà acquistare un calcolatore che offra le stesse prestazioni a metà prezzo. Un esame della stabilità dei costi dell'hardware e del software dimostrerà la consistenza di questa opinione!

Il costo del software di un calcolatore, come si è detto, comprende e quello dei programmi applicativi, e quello della programmazione presso l'utente. Benché ci siano molti programmi applicativi a basso costo, si tratta in genere di programmi con possibilità limitate. Il prezzo del software con alte prestazioni, essenziale ai fini della produzione di un alto reddito differenziale, scende molto lentamente; e, anche se si dimezzasse da un giorno all'altro, il risparmio che ne deriverebbe sarebbe solo dal 5%÷10% sul costo del sistema.

Non solo, ma il prezzo del software realizzato presso l'utente stesso sale abbastanza rapidamente, a causa dell'inflazione, dell'esplosione della domanda, e del diminuire della disponibilità di programmatori.

I costi dell'hardware del sistema comprendono i costi delle periferiche, come il CRT, la stampante e i floppy disks, ed il costo del calcolatore vero e proprio. È bene osservare che negli ultimi tre anni i prezzi di quasi tutte le periferiche si sono mantenuti invariati, o sono aumentati, e che solo le schede di memoria e di CPU hanno subito un abbassamento di prezzo significativo. Questi componenti, però, rappresentano una parte talmente piccola di tutto il sistema, che anche un dimezzamento dei loro prezzi porterebbe ad un risparmio solo del 3% sul prezzo globale del sistema. Le periferiche e lo châssis portaschede comportano la costruzione meccanica, parti mobili, e grandi quantità di componenti elettronici convenzionali; tutte cose soggette all'inflazione, e che non beneficieranno in misura sostanziale per i prossimi anni del tipo di microelettronica utilizzata nelle calcolatrici tascabili e nelle schede di memoria o di CPU.

Si parla molto della nuova tecnologia delle bolle magnetiche e dei dispositivi CCD, ma questi non sono fattori tali da ridurre i costi dei sistemi gestionali per piccole aziende. Con riferimento alla memoria di lavoro, si hanno soltanto degli effetti minimi sui costi, come si è già detto. Per quanto concerne il costo delle memorie di massa, queste due tecnologie non lo ridurranno in una misura significativa, in quanto entrambe hanno un costo per "byte" maggiore di un floppy disk completo di driver, e la situazione non cambierà per parecchi anni. Inoltre, anche se avranno dei costi competitivi, non si sostituiranno tanto presto ai floppy disks. Le memorie CCD non sono in grado di assicurare una memorizzazione permanente senza alimentazione, e né le CCD né le bolle magnetiche permettono la sostituzione e l'archiviazione del supporto, cosa che invece avviene con il floppy disk (che può essere caricato come un normale disco). In altre parole, queste tecnologie non sono in grado di fornire una memorizzazione a lungo termine al prezzo di 30 dollari per un milione di "bytes", come il floppy disk (esclusa la parte driver).

Appare chiaro quindi che il costo di un sistema gestionale completo si manterrà stabile per qualche anno, e che l'eventualità che i prezzi scendano a breve termine è una previsione non realistica, che non deve intervenire nell'analisi del *quando* acquistare un calcolatore. E, cosa forse ancora più importante, anche nell'ipotesi che i costi di un calcolatore dovessero scendere della metà nei prossimi mesi, questo ritardo avrebbe un costo da cinque a dieci volte più alto

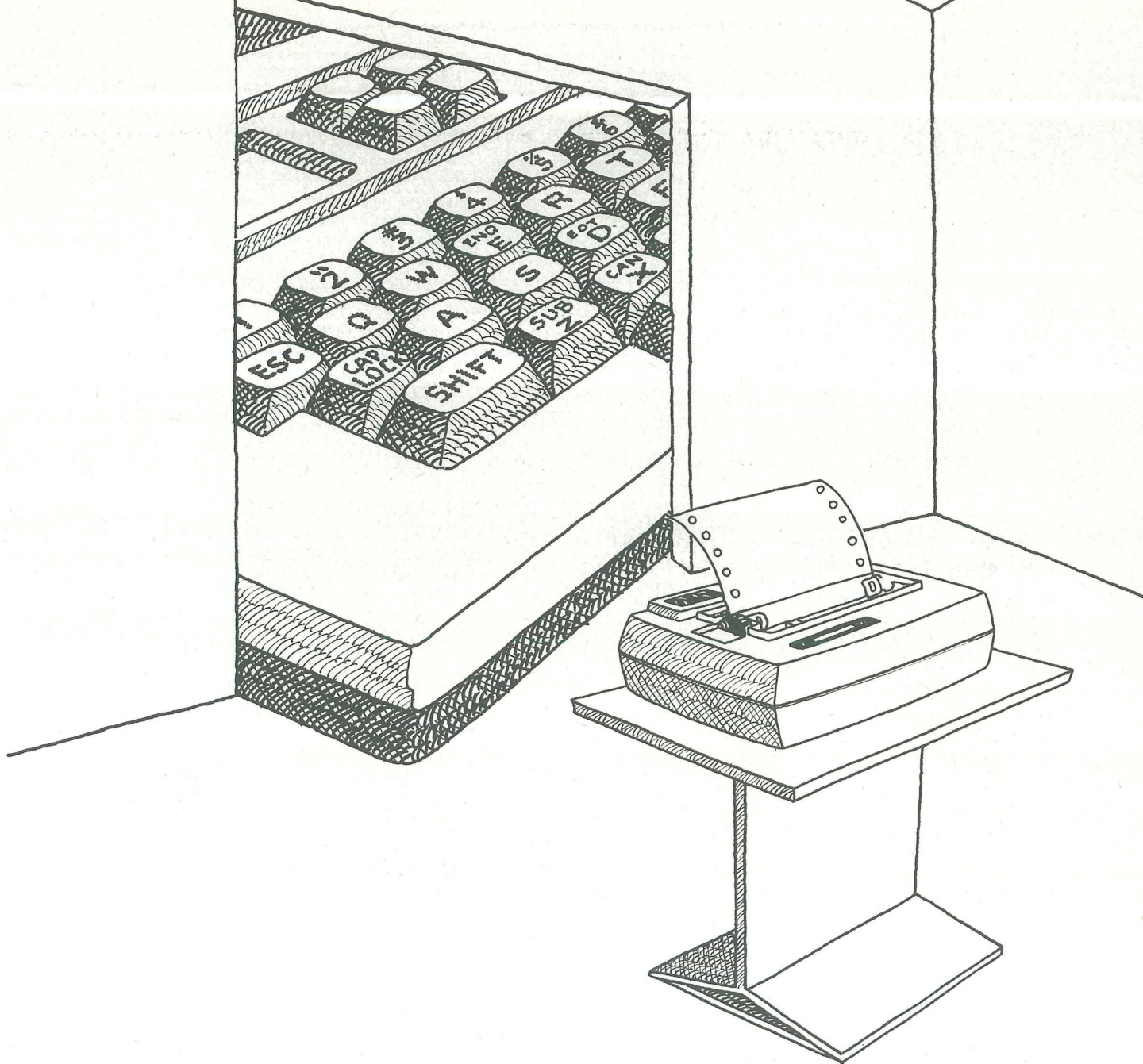
dell'eventuale risparmio, per di più in modo permanente ed irreversibile, come si è sottilineato nel discorso già fatto sull'analisi del movimento di cassa differenziale.
A questo punto, chiarito l'aspetto del costo dell'acquisizione di un calcolatore (uscita diferenziale), restano da illustrare i vantaggi economici che ne derivano (entrata differenziale).

I vantaggi della implementazione di un calcolatore

Il vantaggio più scontato dell'installazione di un calcolatore è la riduzione del costo del lavoro, per quanto riguarda le operazioni di fatturazione, contabilità generale e controllo dell'inventario. Il calcoltore eviterà di ripetere le stesse registrazioni su più di un registro, e la noia di eseguire calcoli e cercare gli errori. Meno evidente è il risparmio derivante dal miglioramento della gestione determinato dal calcolatore. Capita che situazioni critiche non vengano alla luce durante i bilanci trimestrali o annuali effettuati con la contabilità manuale; il calcolatore invece fornisce all'istante i dati sull'attività dell'azienda e dei parametri calcolati. Molto probabilmente da tutto questo si avrà una utilizzazione migliore dal 5% al 10% del capitale investito, che porterà come conseguenza un aumento dal 2% al 4% del profitto lordo.
Oltre a dare un quadro completo delle transazioni correnti, il calcolatore è in grado di elaborare delle proiezioni per il futuro sulla partita doppia (dare e avere), sul costo del capitale, sul movimento di cassa, sui bilanci preventivi, sul controllo dell'inventario e sui parametri relativi all'attività aziendale che presentino dei valori critici. Un ottimo esempio potrebbe essere proprio il tipo di analisi differenziale di cui ci si è serviti nella nostra esposizione, applicabile ad una qualsiasi scelta aziendale e non solo al caso dell'acquisto di un calcolatore.
Un altro servizio, prezioso soprattutto in attività aziendali complesse, è l'analisi dell'andamento critico relativo a processi altamente interdipendenti, quale ad esempio quello della produzione. In assenza di tali strumenti, un'azienda può trovarsi in una situazione di debolezza di fronte a processi critici che non siano stati perfezionati, e che porteranno di conseguenza grossi problemi di consegne e di movimento di cassa.
Gli introiti lordi possono aumentare in modo diretto utilizzando il calcolatore per migliorare le vendite. Si può svolgere un'analisi sulle campagne pubblicitarie per renderle più efficaci; si possono impostare programmi motivazionali per il personale addetto alle vendite, in una maniera che sarebbe impossibile senza il calcolatore, soprattutto per quanto concerne l'analisi dei risultati e degli immediati provvedimenti da prendere. Si può riservare ai clienti un'attenzione personalizzata (ad esempio contatti periodici), riferendosi a dati registrati ed analizzati mediante il calcolatore. Con questi interventi si potrà avere senz'altro un incremento degli introiti lordi dall'1% al 5%, con analogo incremento dei profitti lordi.
Un risparmio non tangibile, ma nondimeno effettivo, si può ottenere facendo elaborare dal calcolatore delle liste di priorità dei lavori, in modo che lavori importanti non vengano procrastinati o lasciati da parte a vantaggio di altri di minore importanza. Rientreran-

no in questo discorso i contatti con i clienti, le commissioni, e tutto ciò che si traduce in denaro se organizzato opportunamente.
Un enorme risparmio potenziale deriva dalla facilità con cui il calcolatore esegue copie della documentazione aziendale più importante, sotto forma sia di dischi magnetici che di stampa, al fine di poterla conservare. La duplicazione di questi documenti non è quasi mai fatta nelle piccole aziende, per cui gli originali sono soggetti ad andar distrutti in caso di incendio, furto, o di altri sinistri. E, quando viene fatta manualmente, è notevolmente costosa, per cui anche qui il calcolatore produce un ulteriore risparmio diretto.
Infine, una fonte di entrata di cui spesso non si tiene conto è costituita dal fornire i servizi del calcolatore a ditte associate, dietro un pagamento forfettario. Ad esempio, spesso un'azienda manifatturiera ha tutta una rete di uffici di rappresentanti, ciascuno funzionante come ditta indipendente. È possibile che le ditte locali desiderino servirsi di un sistema già esistente, soprattutto quando ci siano già dei rapporti di lavoro consolidati e/o nel caso che esse stesse non abbiano la possibilità di investire in un calcolatore proprio.
Questa estensione del calcolatore ad utenti esterni, se è attuata su piccola scala, di fatto è solo un mezzo per utilizzare i tempi morti del calcolatore, e non richiede ulteriori investimenti. Se poi si vuole ricavarne un reddito, si può benissimo portare il calcolatore e prestazioni multiutente e multitask nella sede stessa dell'azienda, con una spesa supplementare molto modesta, dato che il 90% di quanto è necessario è stato già implementato.
Questi vantaggi economici portati dalla computerizzazione rappresentano l'entrata differenziale, che, combinata con l'uscita differenziale di cui si è detto prima, forma la base per una rudimentale analisi del movimento di cassa differenziale, da cui si può partire per valutare la convenienza e la fattibilità della computerizzazione per una qualunque piccola impresa.

Sommario

Riassumendo, la questione di quando dotare l'azienda di un calcolatore si pone come un problema della massima importanza, che, se non è preso in considerazione, può portare come conseguenza dei passivi di cassa molto gravosi, e per di più permanenti, in rapporto a quello che potrebbe essere, cioè in rapporto al potenziale dell'impresa. I costi di acquisizione di un calcolatore sono ragionevolmente dello stesso ordine di grandezza per tutte le piccole imprese. I vantaggi economici illustrati hanno delle grosse potenzialità di estensione, e vanno valutati caso per caso. L'analisi del movimento di cassa differenziale è uno strumento che si può usare rapidamente al fine di ottenere un'indicazione sugli effetti della computerizzazione, sia dal punto di vista della immediata fattibilità che da quello dei risultati di cassa a lungo termine.
La parte II si serve di questi concetti per analizzare un'impresa ipotetica in dettaglio, con un'analisi differenziale più precisa, che tiene conto cioè del costo del denaro, della remunerazione dell'investimento, del deprezzamento, dell'inflazione e del fattore fiscale.

dove la grande stampante non può entrare...

Nuova Honeywell S10 80 colonne la dimensione vincente

• Tecnologia matrice a impatto • 80 CPS con stampa bidirezionale
• Interfaccia seriale fino a 9600 bps e parallela • Alta qualità di stampa
• Self test • Set di 64/96 caratteri

S10: LA SOLUZIONE VINCENTE PER APPLICAZIONI CON MINI, MICRO E PROFESSIONAL PERSONAL COMPUTER

Honeywell

Honeywell Information Systems Italia

Filiale OEM prodotti H.I.S.I. - Via Tazzoli 6, 20154 Milano
Tel. (02) 6570312 - 69771 - Telex 311308 HISI

HARDWARE

8086, Z8000, MC68000: un salto qualitativo nel campo dei microprocessori

di R. Brunelli

Premessa

La realizzazione di microprocessori a 16 bits si propone di dare una risposta, in termini applicativi, alla bassa potenza dei micro ad 8 bits ed all'alto costo dei minicomputers sinora realizzati.
È logico pertanto che su questa strada si siano indirizzati sia i costruttori di semiconduttori che i produttori tradizionali di minicomputers, coll'obiettivo, forse in tempi brevi eccessivamente ambizioso, di realizzare sistemi con alta potenza di calcolo, struttura hardware lineare e flessibile, corredo software sofisticato e completo, basso costo. Vi è comunque per i costruttori di semiconduttori sicuramente l'obiettivo immediato di aumentare la potenza e la velocità degli attuali sistemi a microprocessore 8 bits, mentre per i produttori di minicomputers, quello di ridurre i costi del prodotto.
I costruttori di semiconduttori, nell'affrontare l'area 16 bits, hanno realizzato componenti con architettura hardware estremamente lineare e potente. La flessibilità di applicazione è molto alta e va da sistemi estremamente ridotti e poco costosi a sistemi notevolmente sofisticati, come ad esempio le strutture multiprocessor a bus indipendenti. Il progetto hardware di questi sistemi è relativamente rapido e semplice, in particolar modo quando tutti gli elementi logici presentino lo stesso livello di integrazione e le stesse caratteristiche di interfaccia.
Per contro manca pressoché totalmente a questi componenti un adeguato supporto per la produzione del software, quali compilatori, traduttori, assemblatori, etc.

Il progettista che per la soluzione di una determinata applicazione si accosti a questi componenti trova perciò, accanto all'indubbio vantaggio di poter realizzare una architettura di prestazioni sufficienti e non ridondanti, lo svantaggio di un impegno non indifferente per la preparazione dei supporti software: non è pertanto sempre possibile godere, sul costo finale del prodotto e soprattutto per piccoli volumi, della economicità potenziale di questi componenti.

Per quanto concerne, invece, i produttori di minielaboratori, l'integrazione della propria tecnologia non ha ancora portato alla realizzazione di prodotti realmente a buon mercato, sufficientemente flessibili per potersi adattare in modo ottimale ad una vasta gamma di applicazioni e quindi di prestazioni; l'integrazione in microcircuiti avviene spesso solo per parti altamente ripetitive nel sistema, non solo, ma la struttura software preesistente impone spesso il mantenimento di strutture hardware che, seppure integrate, potrebbero essere migliorabili.

Per concludere si può dire che nel momento attuale la scelta di procedere all'implementazione di un sistema con l'utilizzo dei nuovi microprocessori a 16 bits è senz'altro valida quando l'applicazione comporti una relativa rigidità di utilizzo, risposte estre-

mamente rapide, alta occupazione di memoria associata a tasks consistenti e numerosi ma lineari, nonché, non dimentichiamolo, quando i volumi di produzione siano sufficientemente vasti. Questa scelta diviene invece molto più discutibile, nei confronti dell'utilizzo di tradizionali strutture di minielaboratori, quando ad una applicazione software complessa e flessibile si associno esigenze di produzione e debugging del software stesso.

8086, Z8000, MC68000

Tre componenti sono emblematici di quanto possa attualmente fornire la produzione di semiconduttori: 8086, Z8001, MC68000.
Lo Z8002, in quanto subset dello Z8001, non sarà descritto in questo articolo.
La descrizione che seguirà è fatta con l'intento di fornire un'idea sufficientemente precisa della collocazione di questi componenti nella attuale realtà sistemistica e non intende fare confronti; non perché questo non sia necessario nel compiere una scelta di progetto, ma perché un confronto ha senso solo in relazione ad applicazioni e prestazioni ben determinate.

Vorrei ricordare come la velocità intrinseca di determinate operazioni può non essere significativa se non viene considerata in relazione all'utilizzo in un determinato set di programmi; così come la potenza intrinseca di una unità centrale può non aver senso se non è corredata da un insieme di devices di controllo I/O, memoria, risorse comuni della stessa potenza; e come componenti strutturalmente molto interessanti non possano essere utilizzati sinché non esistano sufficienti garanzie di second sources.

8086, Z8001, MC68000 sono realizzati con tecnologia HMOS ed hanno occupazione 40, 48, 64 pins rispettivamente; l'interfaccia dati ha parallelismo 16 bits, quella indirizzi è di 20, 23, 23 bits e determina la quantità di posizioni memoria visibili in modo diretto dal componente.

L'MC68000, diversamente dagli altri due, presenta queste interfacce già demultiplexate e dotate di logica arbitro per il pilotaggio di risorse comuni (bus o memoria), al fine di ottimizzare la flessibilità del componente e di incrementare al massimo il through-put del bus.
La disposizione funzionale dei pins, per i tre componenti, è indicata in Figura 1.

Struttura dei registri

La struttura interna dei registri è un elemento determinante per la flessibilità e la potenza di calcolo del componente.

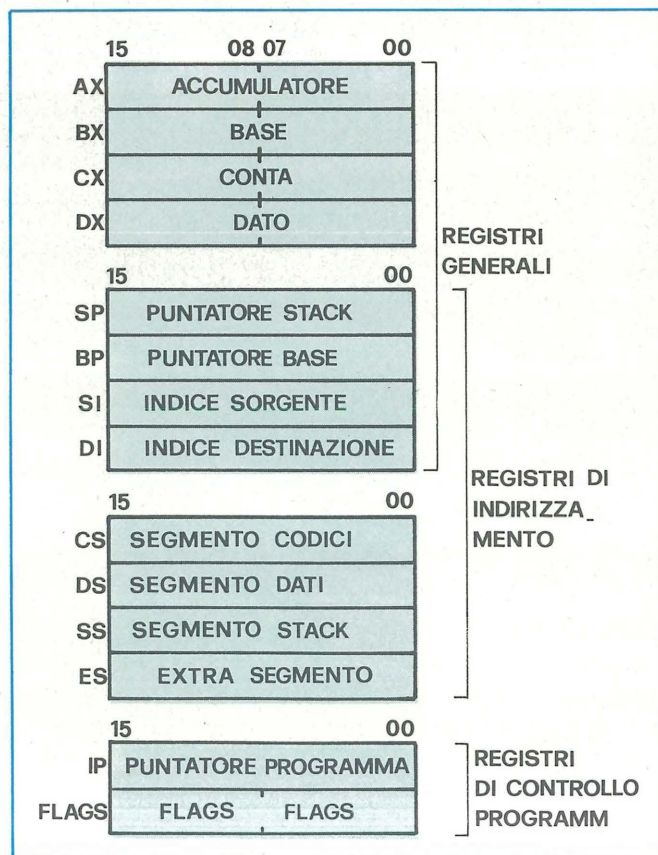



Figura 2 - Struttura registri interni per 8086. È composta da tre gruppi di quattro registri, con parallelismo 16 bits: gruppo registri generali, gruppo registri pointer e index, gruppo registri segmento. In aggiunta esistono i registri per puntatore di programma e flags.

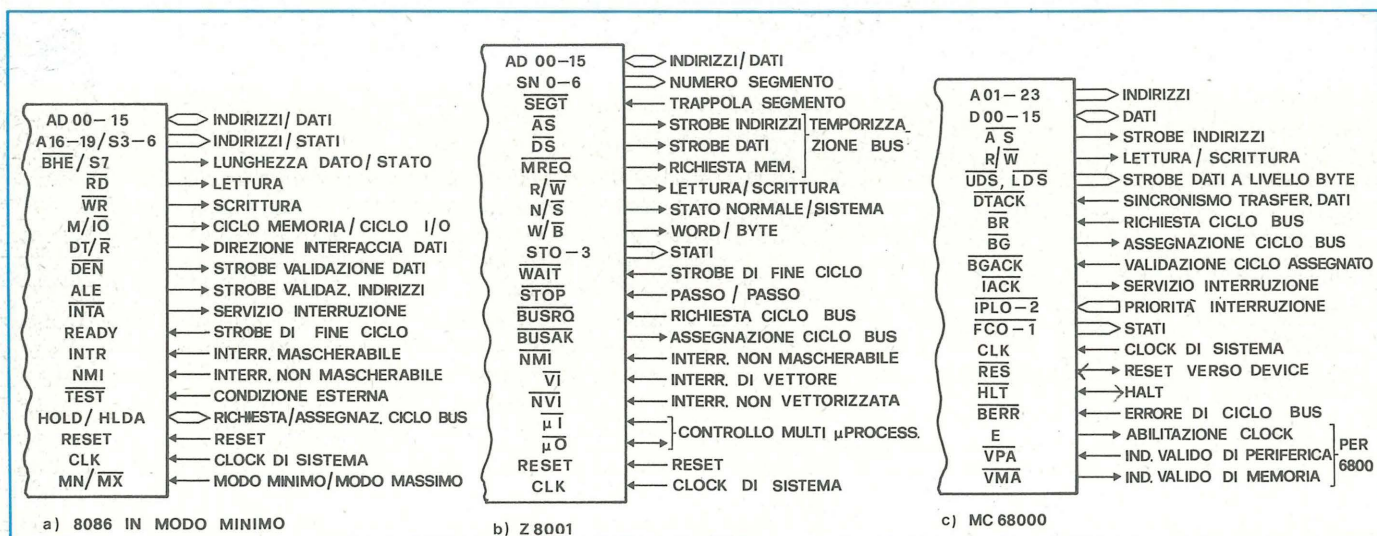



Figura 1 - Disposizione funzionale dei pins di interfaccia per 8086, Z8001, MC68000. Dati e indirizzi sono multiplexer per 8086 e Z8001, demultiplexati per MC68000.

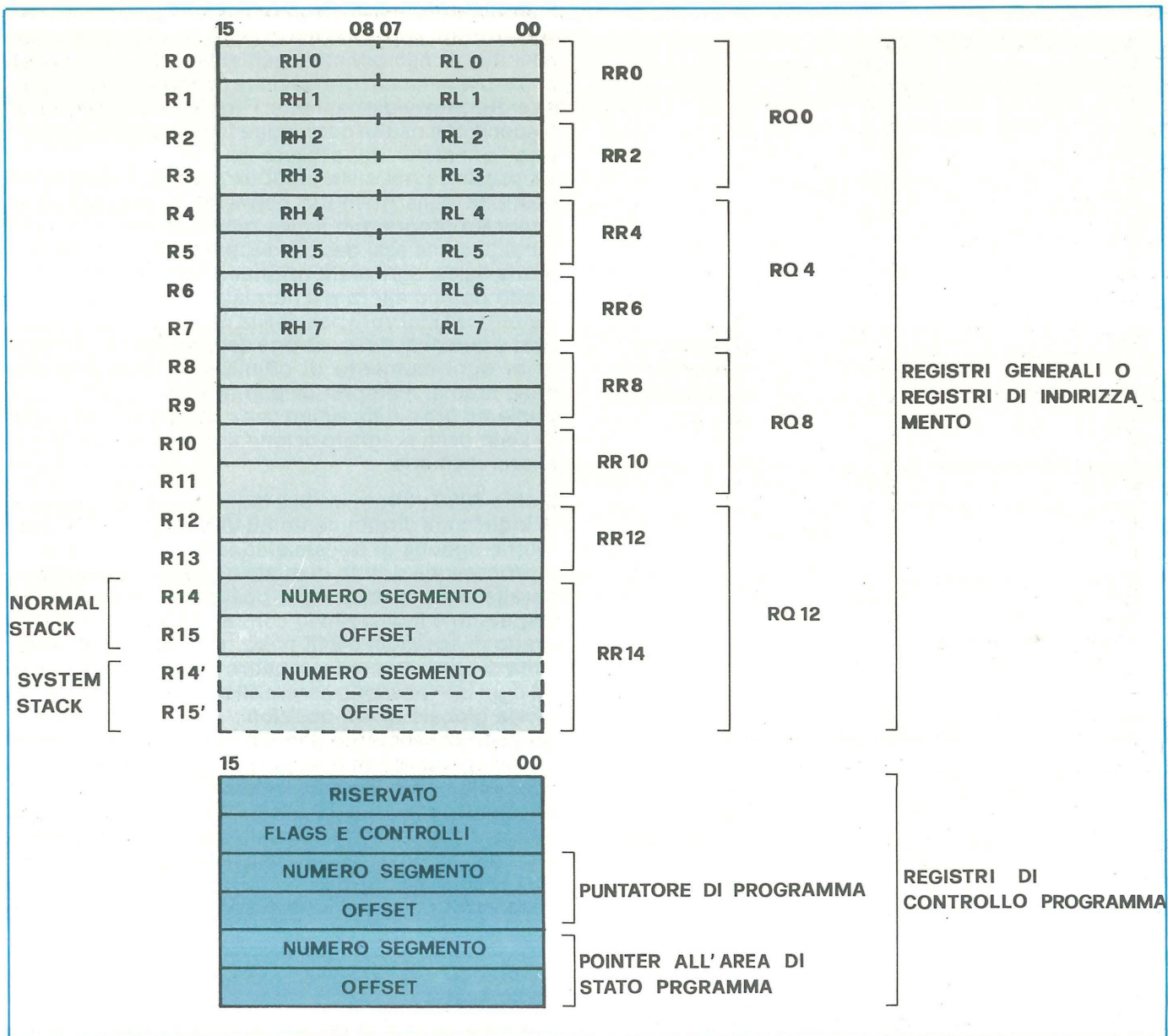



Figura 3 - Struttura registri interni per Z8001. È composta da 16 registri generali di parallelismo 16 bits e da un set di registri speciali di sistema. Tutti i registri generali possono essere usati come accumulatori, ognuno di essi come registro indice o pointer di memoria. La flessibilità è realizzata raggruppando e sovrapponendo tra loro i vari registri.

Il set di registri disponibile in questi componenti può essere raggruppato in tre categorie:

- *registri generali:* sono usati come memoria di lavoro per tutte le operazioni di calcolo interno
- *registri di indirizzamento:* utilizzati per il calcolo degli indirizzi di memoria; questo set di registri può essere comprensivo di registri segmento per il management delle risorse di memoria
- *registri di controllo programma:* contenenti l'indirizzatore di programma e lo stato di questo.

Le Figure 2, 3, 4 illustrano la struttura registri dei tre componenti in oggetto.

Modi di indirizzamento

Si intendono per modi di indirizzamento le possibilità di composizione dell'indirizzo visibili a chi programma per l'accesso all'area dati.
Ogni istruzione fornisce questa indicazione in modo esplicito od implicito.

I modi di indirizzamento disponibili sono indicati di seguito e dettagliati in Figura 5.

- *assoluto:* l'indirizzo di memoria è contenuto nel corpo dell'istruzione
- *di registro*: il nome del registro è indicato dalla istruzione
- *indiretto da registro:* l'indirizzo è il contenuto di un registro indicato dalla istruzione
- *indiretto base o index + displacement:* il displacement contenuto nell'istruzione è aggiornato con il contenuto di un registro base o index
- *indiretto base + index:* l'indirizzo è calcolato come somma di un registro base e di un registro indice, il cui nome è contenuto nell'istruzione
- *indiretto base + index + displacement:* il displacement contenuto nell'istruzione è aggiornato con i contenuti di un registro base e di un registro indice
- *indiretto con aggiornamento:* l'indirizzo è contenuto in un registro e viene aggiornato prima oppure dopo l'operazione di memoria

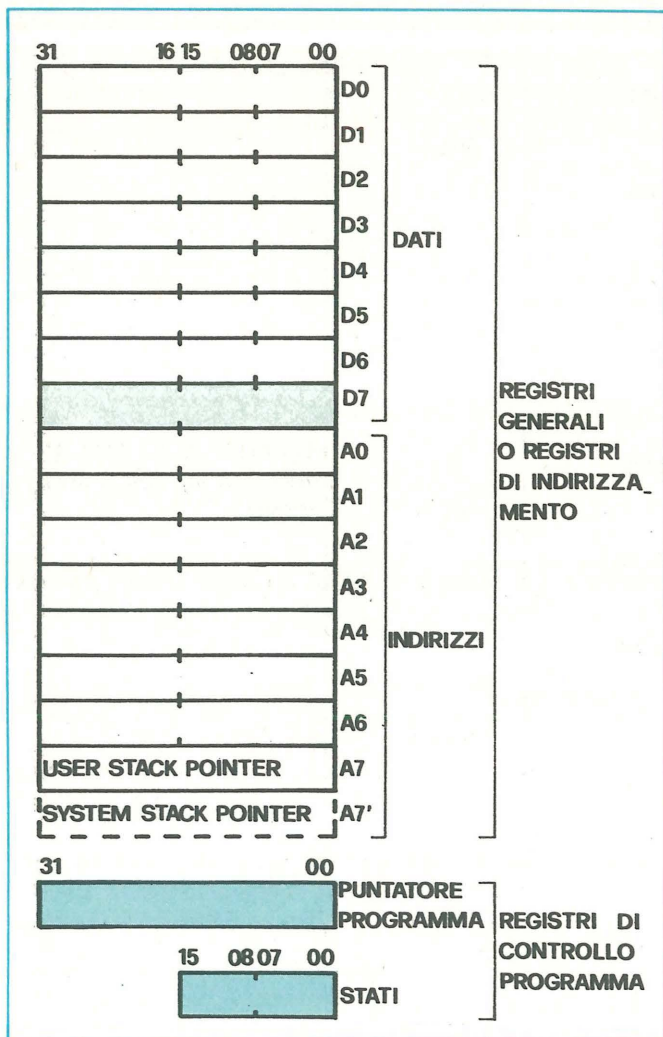



Figura 4 - Struttura registri interni per MC68000. È composta da 8 registri dati, da 8 registri indirizzi di parallelismo 32 bits, oltre a puntatore di programma e registro di stato.

- *immediato:* il dato è contenuto nell'istruzione
- *relativo:* il contenuto del contatore di programma è aggiornato della quantità specificata nell'istruzione
- *relativo con displacement:* il contenuto del contatore di programma è aggiornato con il contenuto di un registro e con la quantità specificata nell'istruzione.

Segmentazione e gestione delle risorse di memoria

L'avvento dei microprocessori a 16 bits indirizza l'applicazione verso soluzioni di multiprogrammazioni con memoria virtuale dove più programmi utente sono attivati simultaneamente.
Questo implica la soluzione di problemi quali la suddivisione dei programmi, l'assegnazione a programmi differenti di risorse comuni oppure dedicate, la gestione di moduli software transienti; non solo, ma è di vitale importanza che un programma non possa inavvertitamente (ad esempio per un errore di codifica) distruggere o comunque alterare un altro programma, sia esso un modulo utente o un modulo di sistema operativo.
L'architettura dei microprocessori considerati prevede questa applicazione in congiunzione con chips specifici di memory management. L'utilizzo di questi chips presenta ancora qualche problema di comparsa e di approvvigionamento; l'implementazione di reti specifiche risulta comunque facilitata dall'architettura "orientata" del processore.
La presenza nel sistema di un'efficace supporto di gestione della memoria permette all'utilizzatore di generare programmi senza minimamente preoccuparsi di come essi debbano essere allocati nella memoria fisica: ogni malfunzionamento è evidenziato in modo da non alterare il normale funzionamento del sistema ed inoltre la possibilità di ricalcolo degli indirizzi e quindi di riallocazione dei programmi consente in ogni momento di ottimizzare l'assegnazione delle aree di memoria disponibili.
Vediamo ora più da vicino i tre componenti dal punto di vista della predisposizione architettonica alla gestione memoria.

Il chip 8086 utilizza un bus indirizzi a 20 bits, capace di indirizzare direttamente 1.048.576 posizioni di memoria; ognuna di queste è considerata di 1 byte.
Il processor è dotato di quattro registri segmento di parallelismo 16 bits; ogni posizione all'interno del segmento è indirizzabile con un offset di 16 bits, in grado di accedere a 64K posizioni, dimensione massima di ogni segmento; i quattro segmenti disponibili ad ogni task possono essere allocati in un'area di memoria globale di 1M posizioni.
I registri di segmento debbono essere caricati da sistema operativo con i parametri propri del task ogniqualvolta ne sia lanciata l'esecuzione. Ogni registro segmento contiene la base, ovvero l'indirizzo più basso di ogni segmento: tutti gli indirizzi del programma saranno aggiornati automaticamente per l'accesso a memoria.
I segmenti a disposizione di ogni programma utente sono quattro:
- *segmento codici:* contiene le istruzioni
- *segmento dati interni:* contiene i dati utilizzati dal solo task
- *segmento dati intertask:* contiene i dati condivisi con altri task
- *segmento stack:* contiene i dati organizzati a stack.

Ogni istruzione indica implicitamente, a seconda del dato su cui opera, quale segmento utilizzare per il calcolo dell'indirizzo; i tre segnali di stato, che caratterizzano ogni accesso al bus di sistema, riflettono questa indicazione.
La Figura 6 illustra quanto descritto.

Il chip Z8001 utilizza un bus indirizzi di parallelismo 23 bits, in grado di accedere direttamente a 8.388.608 posizioni di memoria, ognuna delle quali è considerata di 8 bits.
I 23 bits di indirizzo sono suddivisi in offset di 16 bits e numero segmento di 7 bits; lo spazio di memoria disponibile è pertanto divisibile in 128 segmenti rilocabili, ognuno di essi con dimensione massima 64 K bytes.
Un indirizzo segmentato può essere contenuto in registri macchina, in memoria, nel corpo di una istruzione e le due parti di cui è composto sono operabili da istruzione.
L'architettura del componente è orientata alla realizzazione di un calcolo indirizzo di questo tipo: ad

Figura 5 - Modi di indirizzamento.

Figura 6 - a) Formazione dell'indirizzo, e b) organizzazione della memoria, per 8086.

ognuno dei 128 segmenti rilocabili viene associato in modo dinamico un indirizzo di base segmento; l'indirizzo fisico di memoria è ottenuto sommando l'offset alla base, come indicato in Figura 7.
L'interfaccia del componente è dotata di quattro segnali di stato, i quali indicano 6 possibili casi di accesso alla memoria: istruzioni, dati, stack, per stato user e per stato sistema.
Questi segnali di stato possono essere usati per indirizzare spazi separati di memoria e potenziare così l'indirizzabilità globale del componente.
L'MC 68000 utilizza un bus indirizzi di parallelismo 23 bits, in grado di accedere direttamente a 8.388.608 posizioni di memoria; ognuna di esse è considerata di 16 bits.
La gestione di memoria per cui il componente è stato predisposto assegna ad ogni programma utente 8 segmenti; ognuno di essi è definito da indirizzo di inizio, indirizzo di fine, offset e da un set di indicatori. Queste informazioni sono caricate da sistema operativo al lancio di ogni nuovo task.
I 16 bits più significativi dell'indirizzo logico uscente dal processore definiscono il segmento e vengono sommati all'offset corrispondente, riconosciuto per confronto.
L'indirizzo fisico così ottenuto viene completato, per i 7 bits meno significativi, con la restante parte dell'indirizzo logico, ed inviato alla memoria.
Per ognuno degli 8 segmenti può essere indicato uno specifico utilizzo, in corrispondenza di quanto indicato dai 3 segnali di stato che caratterizzano ogni operazione del microprocessore (area dati, area stack, area codici, etc.).
La dimensione di ogni segmento è variabile da un minimo di 256 bytes ad un massimo che è tutta la memoria indirizzabile, ovvero 16 M bytes.
La procedura di formazione indirizzo è indicata in Figura 8.

Si è cercato di evidenziare con questa breve descrizione l'orientamento seguito dai costruttori nel definire l'architettura del processore ai fini della gestione di memoria.
I due componenti Z8000 e MC68000 sono potenzialmente orientati ad applicazioni più vaste, ma necessitano di un chip aggiuntivo specifico per la realizzazione anche parziale di funzioni di gestione memoria.
L'8086 è meno potente e quindi meno aggressivo verso strutture mini sofisticate ma ha il pregio di contenere hardware dedicato alla gestione dinamica della memoria. Questa scelta è realistica ed equilibrata ed ottimizza il progetto, sia dal punto di vista dell'integrazione che da quello della stesura e della sicurezza del software, per tutta l'area di applicazioni che rientrano nella potenza del componente.
Vorrei ricordare ancora come la struttura di questi processori consenta di impostare con facilità modelli diversi di gestione e protezione della memoria, qualora si voglia rinunciare al livello di integrazione offerta da chips specifici.

Set di istruzioni

Il set di istruzioni disponibili, al di là della indicazione numerica fornita dai costruttori, si articola su cinque gruppi fondamentali:
- *movimento dati* (v. Tabella I)
- *operazioni logiche e aritmetiche* (v. Tabella II)
- *rotazione e shift* (v. Tabella III)
- *operazioni sul bit* (v. Tabella IV)
- *controllo programma* (v. Tabella V)

Le istruzioni di movimento stringa sono di particolare interesse, in quanto permettono di trattare con una sola istruzione gruppi di dati che altrimenti ri-

chiederebbero loops software di una certa dimensione.
Sul chip 8086 il trattamento di stringhe è ottenuto facendo precedere al codice operativo di ogni specifica istruzione un prefisso di ripetizione indicante in modo indiretto la lunghezza dell'operazione.
Sugli altri due componenti l'informazione di ripetizione è contenuta direttamente nel codice operativo, come nello Z8000, oppure è demandata a particolari istruzioni di salto, come sull'MC68000.
I possibili parallelismi sono: byte o word per 8086, byte, word, long word per Z8000 e MC68000.

Le istruzioni di somma e sottrazione possono essere eseguite con un peso massimo word per 8086, long word per Z8000 e MC68000.
MC68000 e 8086 offrono la possibilità di operare memoria e registro in memoria.
Le istruzioni di moltiplicazione accettano un peso massimo di: word operandi, long word risultato per 8086, MC68000; long word operandi, double long word risultato per Z8000.
Le istruzioni di divisione accettano un peso massimo di: long word/word operandi, long word risultato per 8086, MC68000; double long word/long word operandi, double long word risultato per Z8000.
È di rilievo la presenza di istruzioni di somma, sottra-

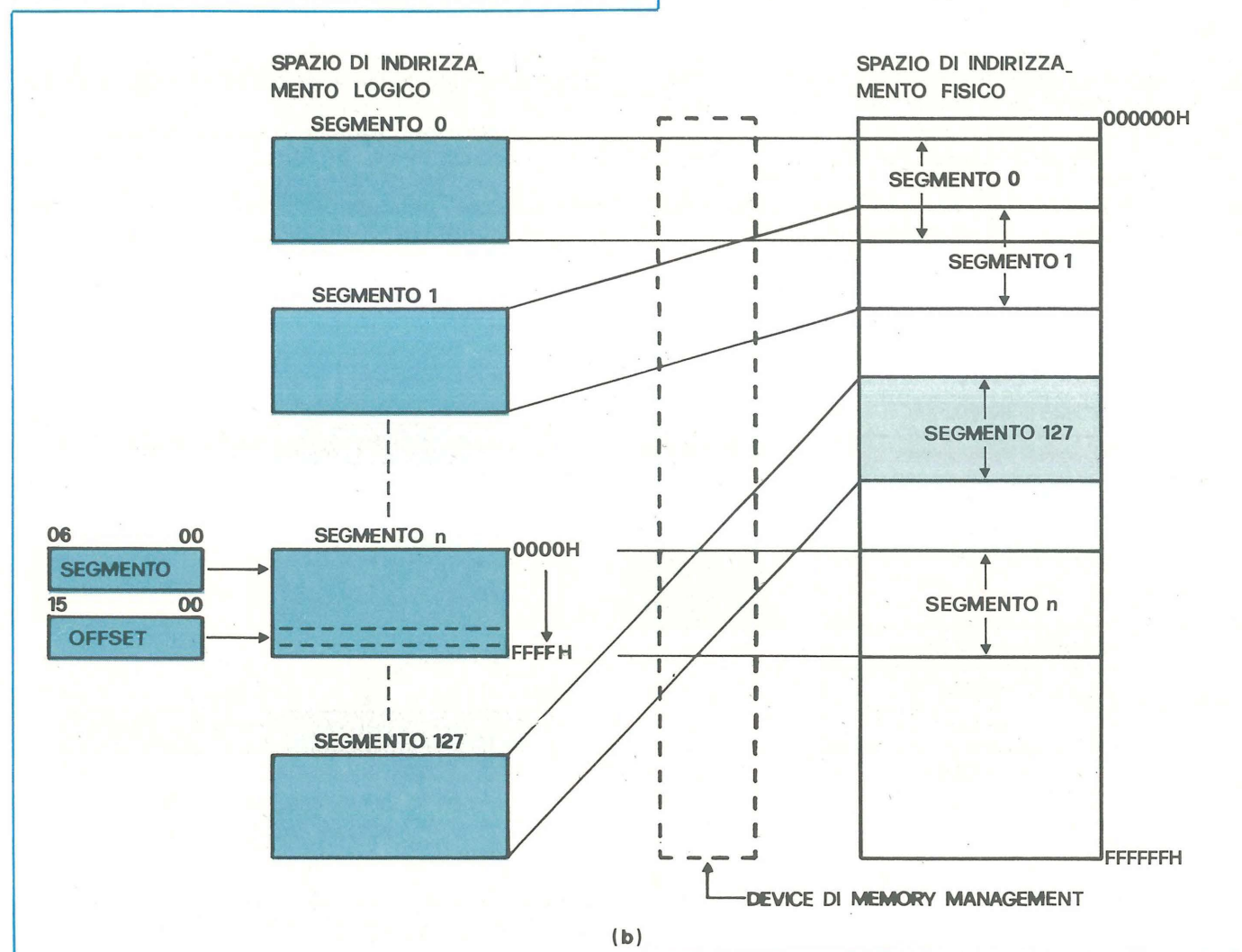
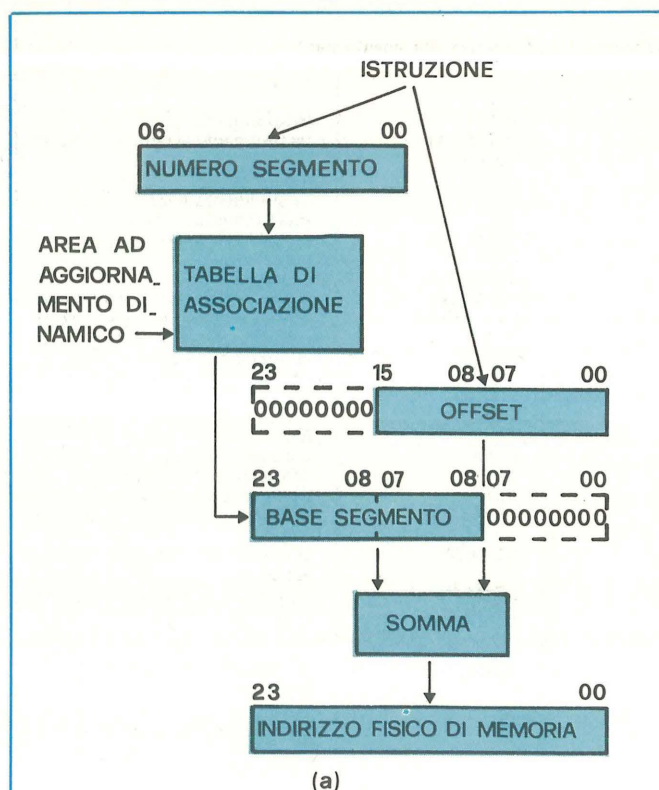



Figura 7 - a) Formazione dell'indirizzo, e b) organizzazione della memoria, per Z8001.

Tabella I Istruzioni di movimento dati					
8086	MOVE	trasferimento	Z8001	LDA,R	caricamento di indirizzo effettivo, assoluto o relativo
	MOVEB,W stringa	trasferimento al byte o word per		LDK	caricamento di costante
	PUSH	trasferimento in stack		LDM	caricamento multiplo
	PUSHF	trasferimento flags in stack		LDR,B,L	caricamento relativo
	POP	trasferimento da stack		PUSH	trasferimento in stack
	POPF	trasferimento flags da stack		POP	trasferimento da stack
	CMP	confronto		CPD,CPI	confronto, incremento o decremento, ripetizione
	XCHG	scambio		CPSD,CPSI	confronto stringhe, incremento o decremento, ripetizione
	IN,INW	input da porta per byte o word		LDD,LDI	caricamento, incremento o decremento, ripetizione
	OUT,OUTW	output a porta per byte o word		TRD,TRI	trasferimento, incremento o decremento, ripetizione
	XLAT	trasferimento per tabella		TRTD,TRTI	trasferimento e test, incremento o decremento, ripetizione
	LEA	caricamento indirizzo effettivo		IN,IND,INI	input da porta, incremento o decremento, ripetizione
	LDS	caricamento pointer in registro DS		OUT,OUTD,OUTI	output a porta, incremento o decremento, ripetizione
	LES	caricamento pointer in registro ES		SIN,SIND,SINI	input speciale da porta, incremento o decremento, ripetizione
	LAHF	caricamento flags in AH		SOUT,SOUTD,SOUTI	output speciale a porta, incremento, decremento, ripetizione
	SAHF	scaricamento AH in flags	MC68000	CLR	azzera operando
	REP	prefisso di ripetizione		EXG	scambio registri
	CMPB,W	confronto di stringa al byte o word		MOVEM	trasferimento registri multipli
	SCAB,W	scansione di stringa al byte o word		MOVEP	trasferimento verso periferica
	LODB,W	caricamento stringa in registro		LEA	caricamento indirizzo effettivo
	STOB,W	caricamento registro in stringa		LINK	caricamento stack
	CLR	azzeramento di registro o memoria		MOVE	trasferimento
	EX	scambio		SWAP	scambio dati nel registro
	LD	caricamento in registro o memoria di operando o di immediato		UNLK	scaricamento stack

zione, negazione direttamente in decimale su MC68000. Non è opportuno approfondire oltre la descrizione di questi componenti: data sheets specifici sono facilmente reperibili ed interpretabili.
Ricordo soltanto alcuni aspetti caratteristici della gestione eccezioni di programma, che trova in questi componenti soluzioni notevolmente più complesse di quelle tipiche per microprocessori a 8 bits.
Le eccezioni, in questa classe di componenti, possono avere origine esterna, come ad esempio interruzioni da device o errore di bus, o interna, come ad esempio istruzioni illegali o trappole particolari.
Queste eccezioni presuppongono la presenza funzionale di uno stato utente e di uno stato supervisore e di hardware specifico per la loro gestione e rimandano a routines specifiche di trattamento individuate

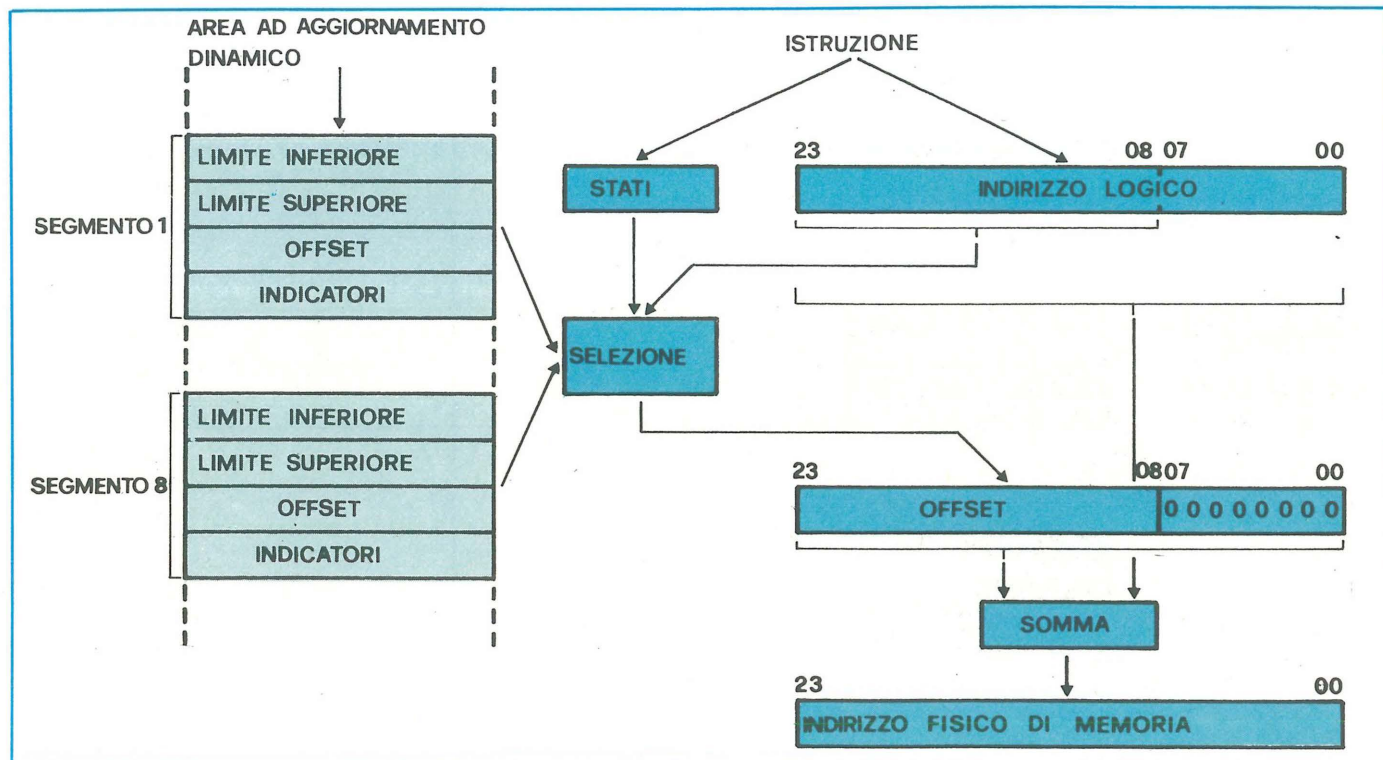



Figura 8 - Formazione dell'indirizzo per MC68000.

Tabella II Operazioni logiche e aritmetiche

8086	ADD	addizione
	ADC	addizione con riporto
	INC	incremento operando
	AAA	correzione BCD per addizione
	DAA	correzione decimale per addizione
	SUB	sottrazione
	SBB	sottrazione con riporto
	DEC	decremento operando
	NEG	negazione
	CMP	confronto
	AAS	correzione BCD per sottrazione
	DAS	correzione decimale per sottrazione
	MUL	moltiplicazione non segnata
	IMUL	moltiplicazione segnata
	AAM	correzione BCD per moltiplicazione
	DIV	divisione non segnata
	IDIV	divisione segnata
	AAD	correzione disimpaccata per divisione
	CBW	conversione da byte a word
	CWD	conversine da word a doppia word
	NOT	inversione
	AND	and
	OR	or
	XOR	or esclusivo
Z8001	ADC	addizione con riporto
	ADD	addizione senza riporto
	CP	confronto con registro o immediato
	DAB	correzione decimale
	DEC	decremento operando
	DIV	divisione
	EXT	estensione del segno
	INC	incremento operando
	MULT	moltiplicazione
	NEG	negazione
	SBC	sottrazione con riporto
	SUB	sottrazione senza riporto
	AND	and
	COM	complemento
	OR	or
	TEST	test operando
	TCC	test conditions codes
	XOR	or esclusivo
MC68000	ABCD	addizione decimale con estensione
	ADD	addizione
	SBCD	sottrazione decimale con estensione
	SUB	sottrazione
	MULS	moltiplicazione segnata
	MULU	moltiplicazione non segnata
	DIVS	divisione segnata
	DIVU	divisione non segnata
	CMP	confonto aritmetico
	AND	and
	EOR	or esclusivo
	EXT	estensione del segno
	NBCD	negazione decimale con estensione
	NEG	complemento a due
	OR	or
	NOT	complemento a uno
	TST	test operando
	TAS	test e set operando
	CHK	verifica dato entro due limiti

attraverso il prelievo di uno specifico vettore di indirizzamento.

Le interruzioni sono caratterizzate ognuna da uno specifico livello di priorità ed intervengono solo se più prioritarie del programma in corso; devono essere mascherabili, ovvero deve essere data possibilità al processore di confrontare le interruzioni in arrivo con un apposito registro, caricato da software: se l'interruzione in arrivo è accettabile viene trattata subito, altrimenti la sua esecuzione viene posticipata. Le funzionalità descritte sono realizzabili con tutti e tre i processori considerati; il livello di integrazione nel chip specifico di CPU è comunque diverso e condiziona la quantità di hardware discreto che è necessario aggiungere.

Si cerca ora di dare un'idea di alcuni aspetti caratteristici dell'implementazione di un sistema. La risposta che i vari costruttori possono dare in proposito, con chips specifici, è un elemento molto importante nel determinare la scelta del componente, in quanto ciò impatta da un lato con l'ottimizzazione del livello di integrazione e quindi con i costi del prodotto, dall'altro con le attività di progetto e prova dell'hardware che possono essere drasticamente ridotte e semplificate, nonché con le caratteristiche di qualità e di durata e con i livelli di potenza dissipata.

Interfaccia tra il componente e gli altri elementi del sistema

È realizzata tramite bus e deve risolvere i problemi di trasmissione e ricezione segnali.
Per i componenti 8086 e Z8000, dotati di unica interfaccia fisica fasata per dati ed indirizzi, occorre realizzare le funzioni di multiplexing, demultiplexing, staticizzazione informazioni, controllo del sincronismo trasmissione-ricezione.

Per sistemi con 8086 le operazioni di controllo e sincronismo sono affidate ad un componente specifico, 8288, il quale pilota i buffer di interfaccia e di demultiplexing.
Per il chip Z8000 non è previsto, per quanto è possibile sapere a tutt'oggi, alcun chip specifico di sincronismo e demultiplexing.

L'unico componente di prossima comparsa è il chip MBU, organizzato a cache memory di capacità 256 bytes; i problemi di demultiplexing e sincronismo sono demandati alla ricezione sui singoli devices.
L'MC68000 è già dotato di interfaccia dati ed indirizzi demultiplexata, con integrazione logica per sincronismo ed arbitro per applicazioni che vedano più di una CPU operante sullo stesso bus.
Le Figure 9, 10 illustrano due possibilità di connessione del chip 8086 al bus di sistema.

Tabella III Rotazione e shift

8086	SHL,SAL	shift logico o aritmetico a sinistra
	SHR,SAR	shift logico o aritmetico a destra
	ROL,ROR	rotazione a destra o a sinistra
	RCL,RCR	rotazione a destra o a sinistra da flag
Z8001	RL,RR	rotazione a destra o a sinistra
	RLC,RRC	rotazione a destra o a sinistra da carry
	SDA,SDL	shift dinamico, logico o aritmetico
	SL,SR	shift aritmetico o logico, a destra o a sinistra
MC68000	ASL,ASR	shift aritmetico a sinistra o a destra
	LSL,LSR	shift logico a sinistra o a destra
	ROTL,ROTR	rotazione senza estensione a destra o a sinistra
	ROTXL,ROTXR	rotazione con estensione, a destra o a sinistra

8284
SEGNALI DI CONTROLLO BUS
8288
8086
8283
INDIRIZZI (00-19)
8287
DATI (00-15)
LOGICA DI SELEZIONE
8287
LOGICA DI SELEZIONE
8287
8259A
INTERRUZIONI (00-07)
8287
8287
MEMORIA O I/O DEVICE
MEMORIA O I/O DEVICE

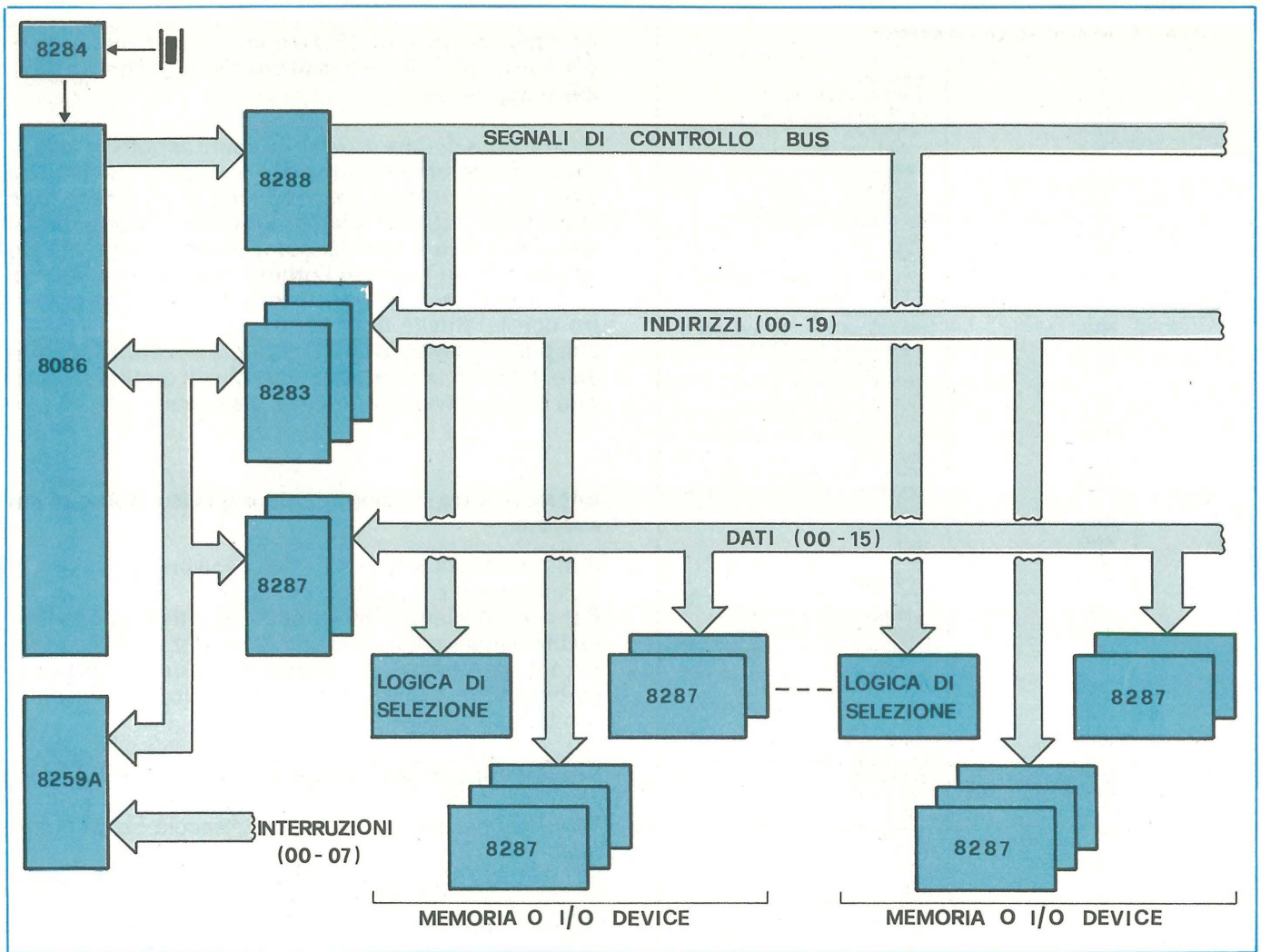



Figura 9 - Connessione della CPU 8086 in modo esteso. Oltre che con il chip 8086 ed il generatore di clock 8284, la connessione è realizzata con l'impiego del bus controller 8288, delle porte di interfaccia 8283 e 8287, dell'interrupt controller 8259A.

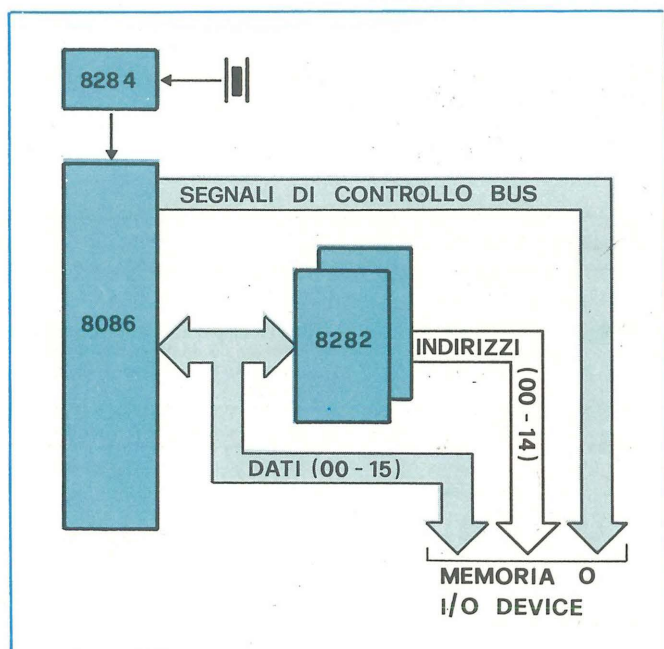



Figura 10 - Connessione della CPU 8086 in modo minimo: un sistema completo, in questa ipotesi, può essere realizzato con 11 componenti, includendo clock, 2 K RAM, 4 K RAM.

Filosofia di dialogo

È la sequenza logica di eventi che regolano lo scambio di informazioni tra due elementi connessi al bus: ad esempio CPU e memoria.
Il dialogo di bus è sincrono per 8086 e Z8000, asincrono per MC68000.

Il dialogo minimo per 8086 è costituito da quattro cicli T1, T2, T3, T4 (v. Figura 11).
L'indirizzo è emesso dal processore durante T1. T2 è utilizzato per l'eventuale cambio direzione delle informazioni sul bus. Il trasferimento dei dati avviene durante i cicli T3, T4. In caso di indisponibilità del dato è inserita tra i cicli T3 e T4 una fase di wait, pilotata dal device.

Tabella IV	Operazioni sul bit	
Z8000	BIT RES SET TSET	test sul bit, statico o dinamico reset sul bit, statico o dinamico set sul bit, statico o dinamico test e set
MC68000	BTST BSET BCHG BCLR	test sul bit test e set test e cambio test e reset

Il dialogo minimo per Z8000 è di tre cicli, T1, T2, T3 (v. Figura 12).
Durante T1 sono presentati gli indirizzi e i comandi sul bus, T2 e T3 sono dedicati alla trasmissione di dati, con eventuale inserzione di cicli di wait.

Sia per 8086 che per Z8000 questi trasferimenti sono eseguiti in modo sincrono, ovvero sono rigidamente associati ai cicli di CPU: il device deve essere sempre in grado di accettare le informazioni di CPU e questa non si fa carico di verificarlo. Solo in caso di devices lenti deve essere inserita in modo fisso la segnalazione di wait. Questa struttura impone di centralizzare la temporizzazione del bus e definisce in modo rigido le caratteristiche di dialogo del device. La CPU non è in grado di riconoscere l'accettazione dei comandi emessi (ad esempio in caso di operazione con la memoria) senza utilizzo della condizione di wait per device inesistente.

La struttura asincrona del bus MC68000 (v. Figura 13) è caratterizzata da un segnale di "informazione presa/pronta" che fa rigidamente parte del protocollo di dialogo e che determina automaticamente l'inserzione o meno di cicli di wait in unità centrale. Questa struttura permette di decentrare la temporizzazione dei vari devices ed è automaticamente sincronizzabile ed ottimizzabile su ogni caratteristica di interfaccia. La CPU è in grado di riconoscere il buon esito dell'invio parametro e di realizzare controlli sul dialogo.

Tabella V Controllo programma

8086	CALL	chiamata
	CLC,D,I	reset riporto, direzione, interruzione
	ESC	delega esecuzione istruzione
	HLT	halt
	INT	alza interruzione
	INTO	alza interruzione se overflow
	IRET	ritorno da interruzione
	JXX	salti su condizione; maggiore, minore, uguale, etc.
	CMC	complemento riporto
	JMP	salto incondizionato
	LOCK	congela assegnazione del bus
	LOOP	gestione di loop incondizionato
	LOOPXX	gestione di loop condizionato
	RET	ritorno da subroutine
	STC,D,I	set riporto, direzione, interruzione
	WAIT	attesa
	TEST	test su condizione esterna
Z8001	CALL	chiamata a subroutine, assoluto
	CALLR	chiamata a subroutine relativa
	DJNZ	decremento e salto se non zero
	IRET	ritorno da interruzione
	RET	ritorno su condizione
	SC	chiamata di sistema
	JP	salto su condizione assoluto
	JR	salto su condizione relativo
	COMFLG	complemento flag
	DI,EI	disabilita o abilita interruzione
	HALT	halt
	LDCTL	caricamento o scaricamento da registro di controllo
	LDCTLB	caricamento o scaricamento da registro flag
	LDPS	carica stato di programma
	MBIT	test bit di multi-micro
	MREQ	richiesta multi-micro
	MRES,SET	reset o set multi-micro
	NOP	non operazione
	RESFLG,SET	reset o set flags
MC68000	BRA,BCC	branch incondizionato, condizionato
	BSR	test di condizione, decrementa e branch
	DBCC	decremento e salto se non zero
	JMP	salto
	JSR	salto o subroutine
	PEA	salva in stack indirizzo effettivo
	RESET	resetta devices esterni
	RTR	ritorno con ripristino
	RTS	ritorno da subroutine
	RTE	ritorno da eccezione
	SCC	set condizionale
	STOP	stop
	TRAP	trappola
	TRAPV	trappola su overflow
	NOP	non operazione

Figura 11 - Temporizzazione di bus per 8086 in caso di lettura memoria.

Multiprocessor: bus arbiter e gestione delle risorse comuni

L'alto livello di integrazione di questo tipo di componenti ed il loro basso costo, associati all'esigenza di ottenere applicazioni in tempo reale, con elevato grado di modularità, porta come tappa obbligata alla realizzazione di strutture multiprocessor a bus indipendenti, comunicanti tra loro con un certo numero di risorse comuni.

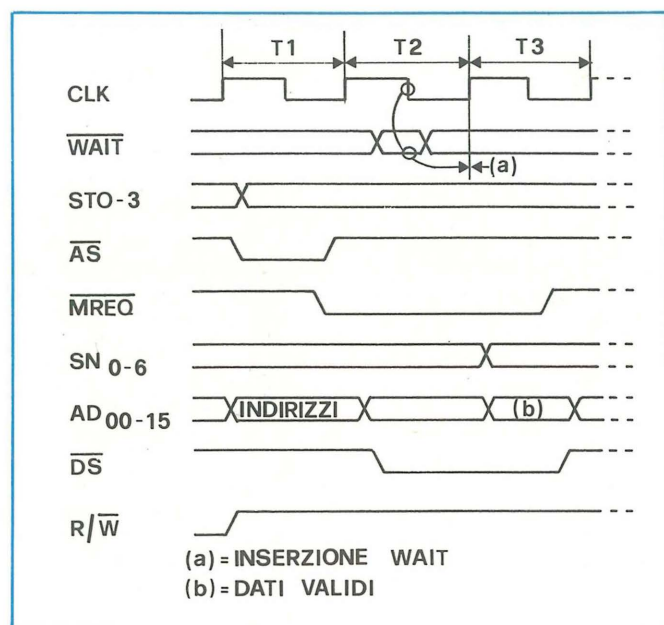



Figura 12 - Temporizzazione di bus per Z8001 in caso di lettura memoria.

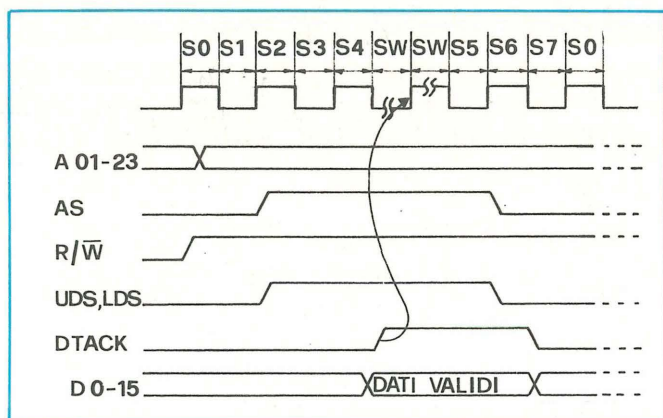



Figura 13 - Temporizzazione di bus per MC68000 in caso di lettura memoria.

Una realizzazione tipica è quella che assegna ad un solo processore l'elaborazione di calcolo interno e dedica a processori specifici il controllo del dialogo con i vari devices; ad esempio: gestione dei disketti, protocollo con le linee, verifiche di coerenza formale, scambio carattere, etc. (v. Figura 14).
Questo tipo di approccio porta il progettista alla suddivisione delle funzioni di sistema in programmi separatamente eseguiti dai singoli processori, con utilizzo concorrenziale delle risorse comuni, hardware e software.
Un primo elemento da affrontare è l'ottimizzazione

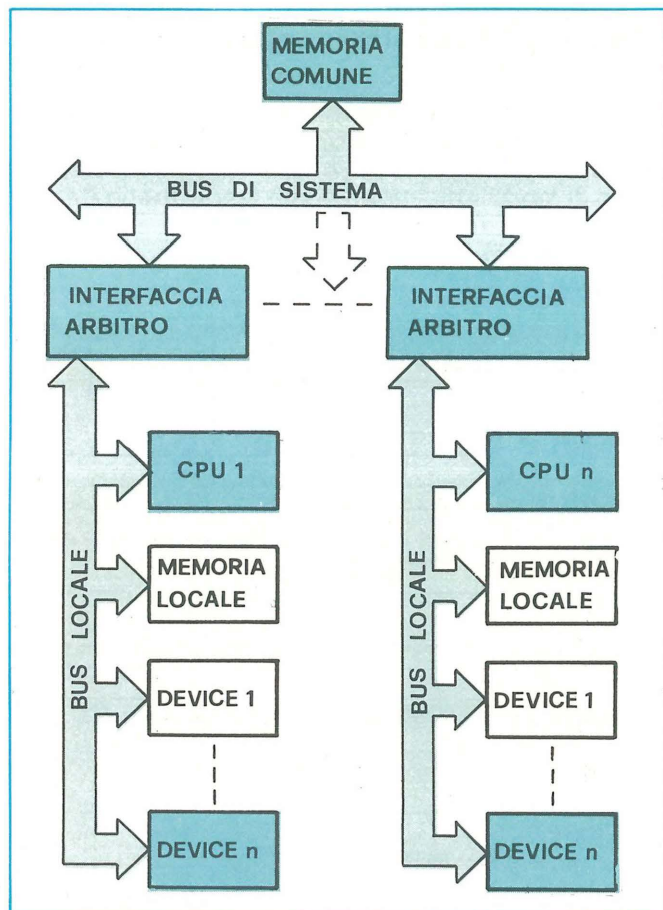



Figura 14 - Struttura semplificata dell'organizzazione a multiprocessor. Le risorse comuni sono gestite in modo concorrenziale dai vari processors locali, specializzati su singole elaborazioni.

delle procedure di scambio informazioni tra i processor e tra processor e device, al fine di migliorare le prestazioni ed evitare sovraccarichi. Un secondo elemento è l'assegnazione di una precisa priorità di intervento per ogni processore, riferita direttamente alle singole esigenze di scambio carattere. Un terzo elemento è la non convenienza, sia in termini implementativi che di prestazioni, del mantenimento di una temporizzazione centralizzata per tutti gli elementi del sistema, i quali devono funzionare in modo asincrono. Un quarto elemento è la determinazione di una logica di dialogo tale da consentire aggiunta o estrazione di elementi dal sistema senza che gli altri debbano in qualsiasi modo esserne implicati.
La soluzione di questi problemi viene affidata dai costruttori a chips specifici, la cui disponibilità a breve scadenza è garantita solo per il bus arbiter 8289 e per l'input/output processor 8089, di supporto all'unità centrale 8086.
Il chip 8289 viene associato ad un bus controller (8288) e ad una CPU (8086 oppure 8089), e gestisce gli accessi al bus comune. Esso è trasparente ai fini della funzionalità di CPU, ed interviene in relazione all'indirizzo di memoria selezionato.
L'8089 è un processore specifico di input/output con capacità di calcolo interno e istruzioni dedicate, specificatamente orientate alla trasmissione di dati in direct memory access tra memoria o device locale e memoria di sistema, con un elevato transfer rate.
Per lo Z8000 non sono stati annunciati a tutt'oggi componenti specifici di bus arbiter; la struttura a multiprocessor appare orientata verso applicazioni caratterizzate da un'unica unità centrale Z8000 con memoria comune di grosse dimensioni e più processori specifici di input/output di parallelismo 8 bits, interfacciati al bus di sistema con buffer di sincronismo. Si tratta del chip UPC (*U*niversal *P*eripheral *C*ontroller) costituito da una versione "device oriented" dello Z80 con memoria interna e logica UART. Il buffer di sincronismo è l'MBU già descritto.
I componenti previsti per MC68000 sono il Bus Arbitration Unit, in grado di effettuare il ricalcolo degli indirizzi per l'accesso alle risorse comuni e il Peripheral Processor Unit.

Sicurezza ed integrità del dato

La struttura multiprocessor, associata alla complessità delle applicazioni, accentua l'esigenza di una tempestiva rilevazione dei malfunzionamenti hardware, sia per garantire un'alta affidabilità dei risultati, sia perché è in molti casi possibile un recupero dell'errore attraverso tecniche di ripetizione o di diversa ripartizione delle risorse di macchina.
Sono fondamentali da questo punto di vista la tempestività della rilevazione e la segnalazione al software del tipo di errore e della parte di macchina in cui si è manifestato.
Si rendono necessarie a questo scopo reti hardware specifiche, in grado di seguire il percorso del dato, con possibilità di intervento a livello di blocchi elementari e di cicli elementari: la struttura globale deve poter disporre di ridondanze specifiche, al fine di poter conservare i parametri iniziali dell'operazione in corso (operandi, indirizzi, lunghezze, etc.).
Appare ovvio a questo punto come sia difficile, qualora queste strutture non siano già state previste a li-

Compucolor II. Grafici a 8 colori, prezzo in B/N.

Non a caso i professionisti si entusiasmeranno di fronte al Compucolor II.

È un sistema completamente integrato, basato sul microprocessore 8080A, con uno schermo grafico da 13 pollici a 8 colori programmabili, con minidisk da 51K per facciata e con l'interfaccia RS232C, il tutto già nella sua versione standard a un prezzo decisamente competitivo.

È programmabile in BASIC, ha 16384 punti indirizzabili sullo schermo e una presentazione di 32 linee per 64 caratteri di testo. La ROM da 16K contenente l'EXTENDED DISK BASIC consente un'accesso casuale ai FILES molto simile allo schema a memoria virtuale tipico dei grandi computers.

Le opzioni del Compucolor II sono costituite da ulteriori FLOPPY DISKS, dall'espansione da 16K a 32K della memoria RAM e da altri 2 tipi di tastiera.

DISTRIBUTORE PER L'ITALIA:
COMPITANT
VIA VITT. EMANUELE III, 9
91021 CAMPOBELLO DI MAZARA (TP)
TEL. (0924) 47153 - (0925) 72325

CONCESSIONARIO PER IL NORD-ITALIA:
SYMIC
MICROCOMPUTERS
& ELECTRONIC SYSTEMS S.R.L.
VIA PONTACCIO 12/a
20121 MILANO
TELEFONO 02/872414

CONCESSIONARIO PER
EMILIA E ROMAGNA, TOSCANA, MARCHE:
SORI S.N.C.
VIA BOLDRINI, 6
BOLOGNA
TELEFONO 051/558311

vello di integrato, procedere a controlli efficaci con reti hardware aggiuntive, essendo praticamente disponibile soltanto l'interfaccia esterna del processore; tanto più quando ci si trovi di fronte all'esecuzione interna di istruzioni che eseguono operazioni attraverso cicli multipli di CPU con ricopertura a livello non solo di registri interni ma anche di memoria principale.
Questo è uno svantaggio non indifferente, se raffrontato alle tecniche decisamente complete di cui possono disporre i minielaboratori, soprattutto per quelle applicazioni in tempo reale e in condizioni ambientali sfavorevoli che possono concedere margini molto ridotti di discontinuità funzionale (si veda il caso ad esempio di centrali di commutazione, di controllo di processi, etc.).
I vantaggi intrinseci all'integrazione di questi componenti vengono in questi casi molto ridotti dalla quantità di hardware discreto che è necessario aggiungere.
Un secondo elemento molto importante è la presenza di strutture autodiagnostiche sul sistema in grado di ridurre drasticamente sia i tempi di intervento per la sostituzione delle parti guaste che il livello di professionalità dell'intervento stesso. Sono necessarie a questo scopo logiche di sezionamento o di ritorno a livello dei vari blocchi funzionali, non previsti nelle strutture di questi chips.
Per concludere si può dire che all'alto grado di protezione raggiungibile con questi componenti pr l'utilizzo del software non corrisponde un'adeguata sicurezza sulla integrità del dato, sulla continuità della elaborazione, sulla facilità di diagnosi e di intervento: tutti aspetti questi che nei minielaboratori hanno già da tempo trovato soluzioni sofisticate ed efficaci.

Conclusioni

Ho voluto dare in queste righe una descrizione panoramica dei tre componenti; la loro potenza è indubbiamente notevole, ma l'utilizzo, in conseguenza delle problematiche di sistema che è necessario risolvere, è quanto mai complesso. Soltanto gli utilizzatori che saranno in grado di compiere il necessario salto qualitativo di progetto, in tempi adeguati, potranno trovare in questi componenti un efficace strumento di conquista del mercato.

COMUNICATO STAMPA

La NEC Electronics Italiana S.r.l. e la Società SILVERSTAR Ltd. S.p.A. hanno concluso un accordo di collaborazione per la distribuzione su tutto il territorio nazionale di tutti i prodotti della linea di componenti elettronici della Nippon Electric, in particolare memorie LSI, nel cui mercato la NEC occupa una posizione di rilievo, transistori plastici, display fluorescenti FIP ed a plasma.

Il dottor Fontana della Microlem sull'ultimo numero di BIT nel suo intervento in questa rubrica, rivolgeva ai lettori l'invito ad "iniziare un dibattito serio, critico e costruttivo" sui temi della professionalità e della cultura tecnologica in Italia. L'invito è stato ampiamente accolto; le lettere che abbiamo ricevuto in risposta testimoniano di un interesse vivo e diffuso su quei temi. E proprio per questa ragione abbiamo pensato di far proseguire il discorso iniziato dal dottor Fontana, dedicando questo numero di Tribuna ad una delle lettere ricevute.

M.M.

Professionalità e cultura scientifica

Fra tanti dibattiti che si fanno oggi in Italia credo che questo sia uno dei pochi costruttivi. Spero che a questo dialogo rivista-pubblico partecipino numerosi i giovani, troppo spesso defilati in problematiche di questo tipo. Ed anche le donne: credo che sarebbe una bella maniera di dimostrare che il femminismo non è solo parolaio e che è finita l'epoca del "tanto-io-di-quelle-cose-lì-non-ci-capisco-nulla".

E riprendiamo il tema sollevato da Fontana: la "professionalità all'italiana". Sa il Cielo se non sono d'accordo con lui: in realtà la situazione è anche più nera di come lui l'ha descritta: oltre a ragazze che non sanno l'alfabeto, esistono medici che non hanno mai visto un cadavere o fatto una sutura; ingegneri che non sanno cos'è un calcolatore; fisici laureati con tesi in elettronica che non hanno mai preso in mano un integrato.

Di chi la colpa? Anzitutto della scuola, dalle Media Inferiori all'Università (in cui lavoro io, detto per inciso!). Ma anche - e forse soprattutto - di un certo tipo di mentalità che vede sempre nella Cultura solamente la Cultura Umanistica. E per di più fatta solo sui libri, dimenticando che il Rinascimento è stato anche un'incredibile rivoluzione tecnologica: il "sapere degli antichi" non era infatti solo poesia o stilistica, ma anche agricoltura ed architettura e conoscenza dei materiali. E tecnologia ad alto livello fu la metallurgia e l'arte del vetro e la ceramica e la liuteria e la pittura: non vuol dire questo disconoscere l'importanza di letteratura, filosofia, diritto ed arte. Vuol dire riconoscere semplicemente che una grossa fetta di Cultura è stata sistematicamente negletta, al punto che oggi una storia tecnologica del Rinascimento deve essere ancora scritta.

Almeno in Italia.

Il guaio è che di tecnologia e scienza oggi il mondo vive: ne ha avuto in regalo (con tutte le limitazioni del caso!) un luogo di periodo di pace ed un progresso senza precedenti nella qualità della vita, tale che oggi (anche nel terzo mondo) nessuno scambierebbe la sua vita con quella di soli vent'anni fa. E le promesse sono ancora più rosee: si dovrà proprio a questo tipo di cultura se l'uomo potrà avere un futuro migliore, ivi compresi quei tre miliardi di persone (una bazzeccola!) che oggi vivono in condizioni difficilmente qualificabili come "umane". E poi è un tipo di cultura che ha un vantaggio: una media intelligenza e - questo si - molto impegno e serietà sono condizioni necessarie e sufficienti perché chiunque possa impadronirsene: in questo senso scienza e tecnologia sono veramente universali.

Quindi tutti gli sforzi possibili vanno fatti per diffondere queste esigenze e stimolare la gente ad interessarsi a questi problemi: con una frase anche troppo abusata potremmo dire che la cultura tecnico-scientifica deve divenire al più presto una vera "cultura di massa". Se fosse così, questo solo fatto farebbe giustizia di molti ciarlatani ed avventurieri.

Il problema è come. Ben vengano le scuole private, ma va detto subito, caro Fontana, che esse da sole non potranno risolvere il problema: potranno infatti rivolgersi a piccoli numeri di utenti, per periodi limitati, mentre ci troveremo ben presto a dover frontaggiare nuove masse di studenti a tutti i livelli, se è vero che - a stare alle statistiche in nostro possesso - gli studenti delle materie scientifiche stanno (udite, udite!) aumentando e sovvertendo tutte le previsioni della vigilia. E quel che è di peggio è che aumentano anche in serietà e voglia di lavorare: è inutile far notare infatti che le scuole oggi possono sopravvivere proprio perché la gente tutto fa fuori che studiare. Guai se tutti studiassero sul serio!

Ben venga anche la scuola pubblica, la grande imputata, in cui però le lungaggini burocratiche, le pastoie amministrative, il rifacimento dei programmi, e l'adeguamento (in tutti i sensi) del personale insegnante non sono certo problemi da poco. Questo vuol dire che anche i tecnici si dovranno mettere a fare politica, anzi una Politica (con la P maiuscola) che sarebbe ora venisse fatta in Italia. Fatta e non discussa.

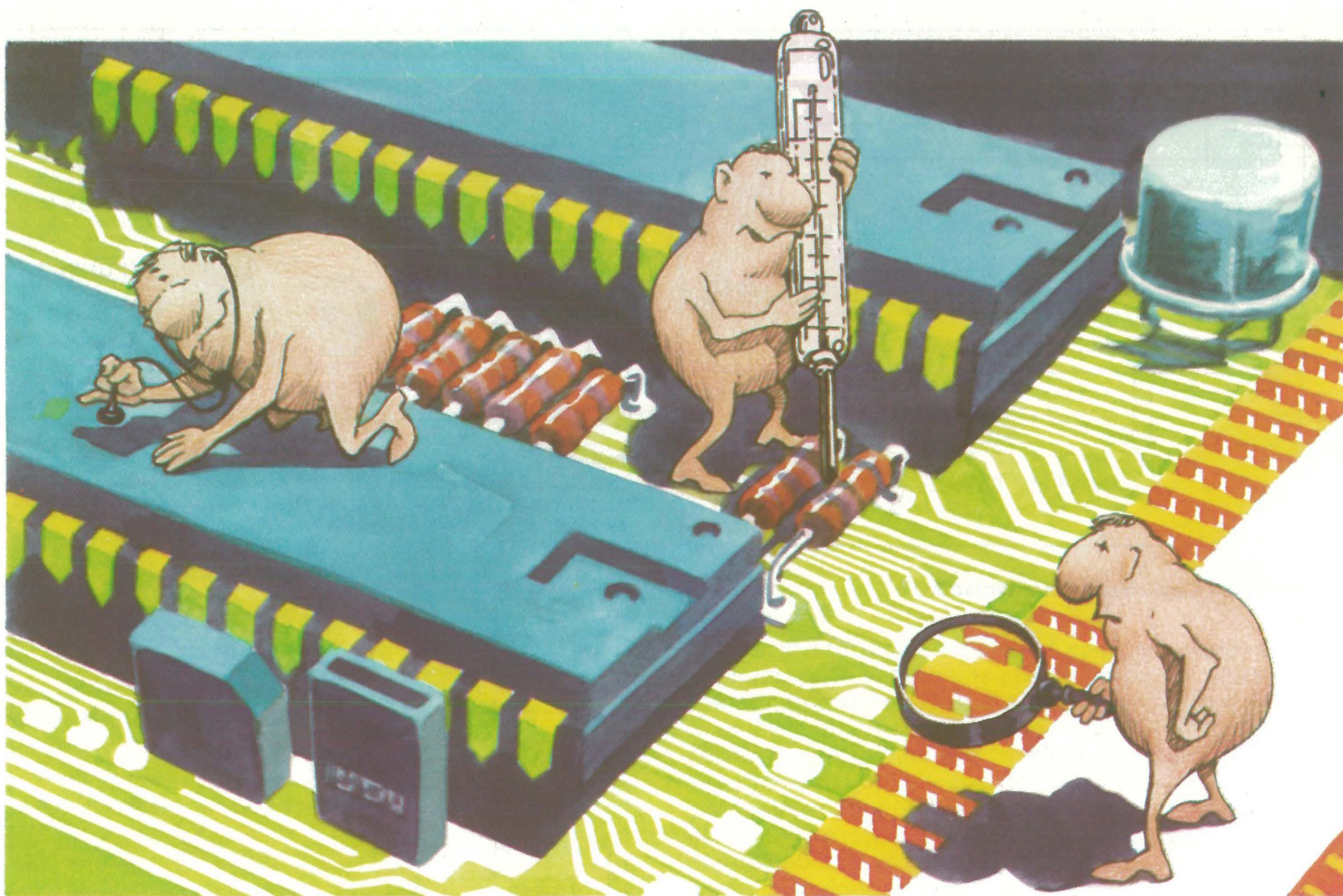
E poi c'è necessità di leteratura scientifica, in Italia. Anche se credo che non sia tanto un problema di Case Editrici, quanto di autori. I libri scritti da italiani per italiani, pur mantenendo una validità a livello internazionale (l'unica che io riconosca valida!) sono troppo pochi. Abbiamo anche troppi Trattati, scritti da gente diabituata da troppo tempo a diffondere la cultura, ed abituata da troppo tempo ad insegnare davanti ad una lavagna (la cosiddetta "Cultura del Gesso") e non nei laboratori.

Ed infine le riviste: credo che le riviste abbiano un ruolo importantissimo da svolgere, proprio per la diffusione che riescono a raggiungere e per la possibilità di avere un continuo ed agile contatto con il pubblico e di muoversi nel senso di quella "continuing education" che è una delle caratteristiche della cultura scientifia: caratteristica che è destinata a divenire sempre più marcata col passare del tempo.

Forse (forse!) con un coordinamento di tutti questi sforzi potremo uscire dal tunnel. La classe politica avrebbe in ciò un gran ruolo da giocare, se non fosse costituita troppo spesso da persone con un livello di conoscenze tecniche a dir poco lacrimevole. E forse ci troveremmo di fronte a discorsi nuovi: radio e TV (private e non), quotidiani e settimanali che cominciano a dedicare alla tecnica qualcosa di più discutibili pezzi di colore: e che cominciano ad insegnare a fare oltre che a sapere: perché è proprio nella dignità di ciò che si conosce e di ciò che si sa fare (e bene!) che consiste la grande differenza fra la cultura scientifica e la pseudo-cultura fatta di chiacchiere. E potremmo anche ragiungere un altro obiettivo: quello di dire chiaro che la cultua tecnico-scientifica è sì aperta a tutti, ma è selettiva, terribilmente selettiva: e non solo per gli studenti, ma soprattutto per gli insegnanti. Troppa gente che insegna poco, male e sbagliato: è meglio essere un cattivo medico, che un cattivo insegnante di matematica.

Caro Dr. Fontana: evidentemente non resta che romboccarvi le maniche, a questo punto. Ma anche con po' di entusiasmo ed ottimismo: siamo in fondo la settima potenza mondiale (incredibile!), e se spingiamo nella stessa direzione, ognuno al suo posto, ognuno per la parte che gli compete, potremo fare parecchio. E credo che ne valga la pena, perché penso che i giovani d'oggi (ragazzi e ragazze) se lo meritino e se lo aspettino: sul serio, subito, e da noi.

Cordialmente

Flavio Waldner

Usiamo il Picocomputer

Parte IV di **D. Del Corso**

"Usare il microprocessore" non vuol dire solo scrivere un programma in BASIC e farlo eseguire. Questo componente propone un nuovo modo di fare l'elettronica anche per cose molto semplici e permette di realizzare, partendo da un unico nucleo base (nel nostro caso il Picocomputer), strumenti, circuiti di controllo, applicazioni di vario genere.
Gli esempi descritti in queste pagine sono stati sceti in modo da soddisfare due criteri:
- l'applicazione richiede interazioni con il mondo esterno non solamente tramite la console di operatore, ma tramite delle interfacce specifiche;
- programmi e periferici possono essere utilizzati come nucleo base per applicazioni più complesse.

Per la loro realizzazione sono sufficienti la scheda Picocomputer ed il Picoperiferico già presentati su BIT (numeri 3, 4 e 5). Possono ovviamente essere usati anche altri Single-Board-Computers, con sostanziali modifiche ed integrazioni dei programmi e delle interfacce qui descritte.
Le prime due applicazioni prevedono una parte hardware ed una parte software; questo per mettere in evidenza la necessaria integrazione di questi due aspetti, caratteristica dell'uso del microprocessore. I programmi proposti sono molto brevi e possono quindi essere assemblati e caricati a mano, usando il Picoperiferico. Anche i circuiti di interfaccia previsti usano un numero minimo di componenti e possono essere montati agevolmente su una basetta per cablaggi senza saldature, collegata al connettore B (rimasto libero), sulla scheda Picocomputer.
I programmi usano anche il visualizzatore e la tastiera; per la loro gestione si sfruttano, ove possibile, le routines del monitor già disponibili (vedi BIT nº 5).

GAME TORINO

a cura del Gruppo Amatoriale Micro-Elaboratori, Torino *e del Laboratorio Didattico di Elettronica Applicata dell'Istituto di Elettronica e Telecomunicazioni del Politecnico di Torino.*

Esempio 1: convertitore D/A a parzializzazione

Obiettivo è controllare da programma una corrente od una tensione analogica, allo scopo di variare la potenza fornita ad un carico (per esempio un piccolo motore o una lampadina). La tecnica usata è quella della parzializzazione, indicata in Figura 1.

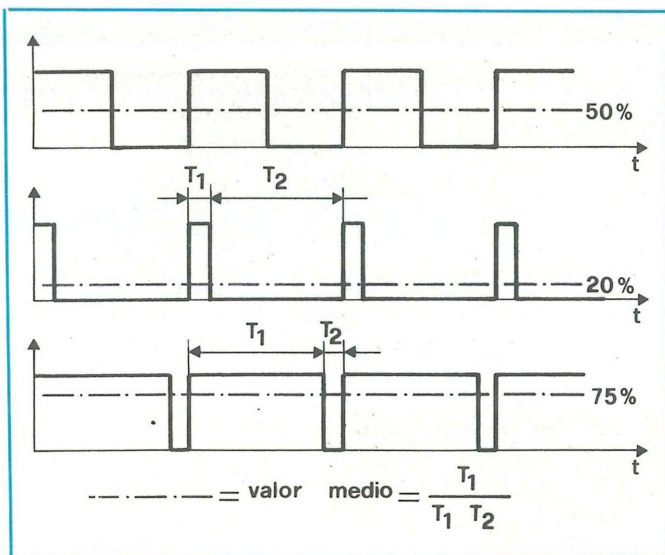



Figura 1 - Tecnica della parzializzazione.

Il problema richiede il progetto di una interfaccia per l'organo "di potenza" e la stesura del programma di gestione. Quest'ultimo a sua volta comprenderà il modulo di controllo vero e proprio ed un modulo di colloquio con l'operatore, che permette di inserire i parametri di controllo. Per quest'ultimo useremo sottoprogrammi già esistenti nel "monitor" residente. Come interfaccia con l'organo attuatore esterno, se ci limitiamo a basse tensioni ed a correnti dell'ordine di qualche centinaio di mA, sono sufficienti normali porte di potenza (buffer) TTL con uscita a collettore aperto. Per non interferire con la gestione del Picoperiferico useremo le due uscite impulsive PUL 1 e PUL 2, rispettivamente come set e reset di un flip-flop che genera direttamente la forma d'onda di comando della parzializzazione. Lo schema dell'interfaccia

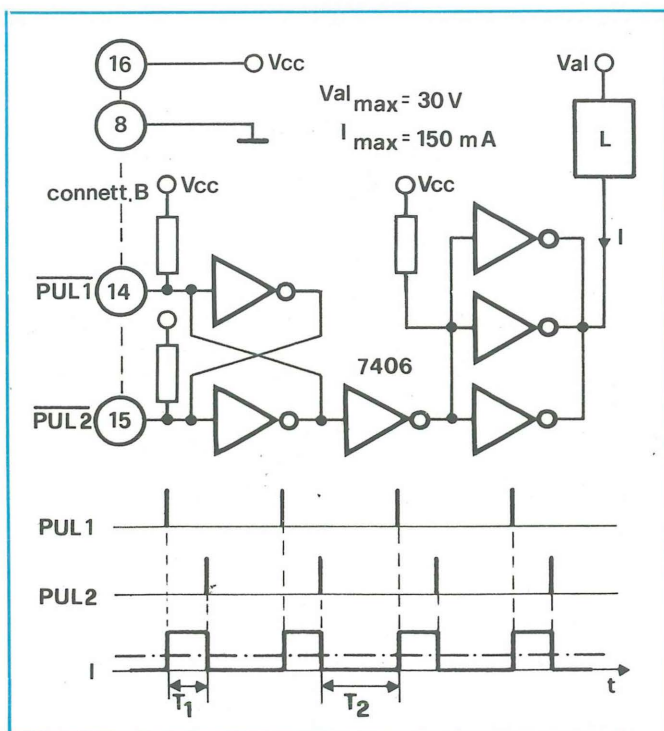



Figura 2 - Interfaccia per controllo di piccoli carichi.

```
CONTR   LD      (PUL1), A
        CALL    DELAY 1
        LD      (PUL 2), A
        CALL    DELAY 2
        JR      CONTR
```

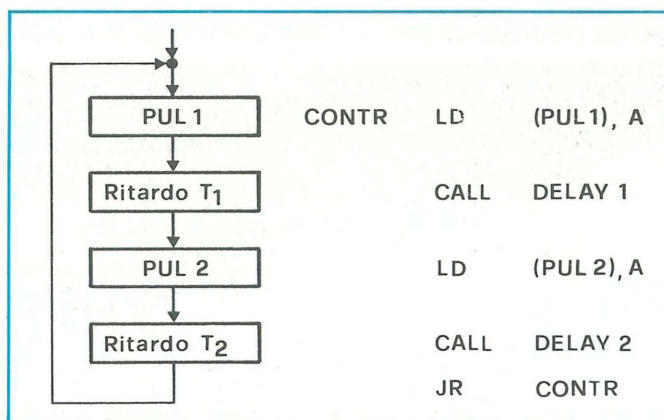



Figura 3 - Controllo della parzializzazione. Le uscite PUL sono attivate da istruzioni di scrittura sugli indirizzi della ROM (v. BIT n. 4). Le routines DELAY1 e DELAY2 sono analoghe alla DELAY del Monitor, ma hanno il parametro variabile.

```
LD      A, (BUFF)
LD      D, A      ; valore T1 in D
LD      A, FFH
SUB     D         ;calcola T2
LD      E, A      ; valore T2 in E
```

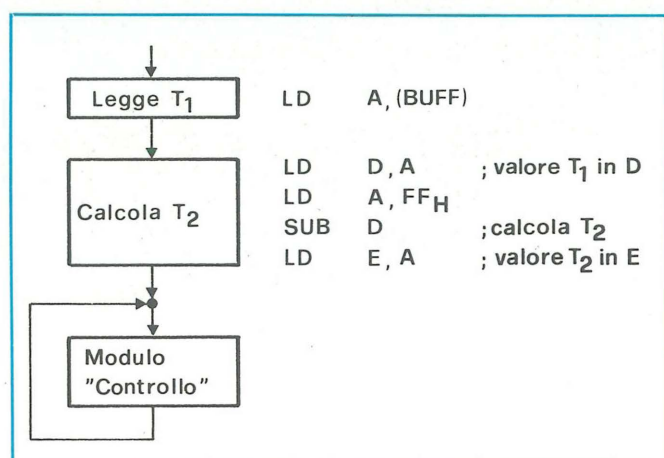



Figura 4 - Controllo con parametro fisso. Il modulo "CONTROLLO" usa il contenuto dei registri D ed E come parametro di DELAY.

esterna è in Figura 2. Il carico L (motore o lampadina) ha costanti di tempo molto maggiori della durata degli impulsi di corrente T_1 e del loro intervallo di ripetizione T_2, pertanto agisce come filtro passa-basso e media la potenza fornita dagli impulsi.
Variando il rapporto T_1/T_2 varia il duty-cycle dell'onda quadra e quindi il valore medio della corrente I nel carico L. La struttura del programma di controllo è molto semplice (v. Figura 3).

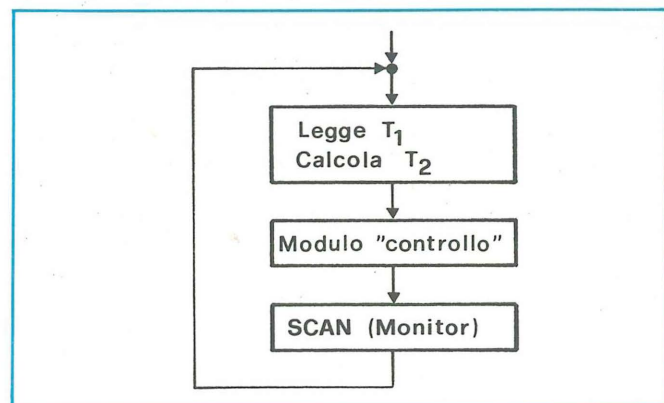



Figura 5 - Controllo con parametro variabile. Il blocco SCAN (presente nel Monitor) aggiorna BUFF e gestisce tastiera e visualizzatore.

```
                         1  ;
                         2  ;
0800                     3            ORG      800H
                         4  ;
                         5  ;
                         6  ;         CONVERTITORE D/A A PARZIALIZZAZIONE
                         7  ;
                         8  ;
                         9  ;
                        10  ;         DEFINIZIONE PARAMETRI
                        11  ;
                        12  BUFF      EQU      0BF0H
                        13  PUL1      EQU      100H
                        14  PUL2      EQU      500H
                        15  COP       EQU      44H
                        16  USP       EQU      0BD0H
                        17  ;
                        18  ;
                        19  ;
0800   31D00B           20  CONV      LD SP,   USP
0803   3AF00B           21  CONV1     LD A,    (BUFF)
0806   57               22            LD D,    A
0807   3EFF             23            LD A,    0FFH
0809   92               24            SUB      D
080A   5F               25            LD E,    A
080B   320001           26            LD (PUL1),A
080E   42               27            LD B,    D
080F   10FE             28  LOP1      DJNZ     LOP1
0811   320005           29            LD (PUL2),A
0814   43               30            LD B,    E
0815   10FE             31  LOP2      DJNZ     LOP2
0817   CD4400           32            CALL     COP
081A   18E7             33            JR       CONV1
                        34  ;
                        35  ;
                        36  ;
                        37  ;
```

LIST 1

Le ultime due cifre introdotte dal campo dati del Picoperiferico vengono poste nella cella BUFF; è quindi comodo stabilire che il valore presente in questa cella rappresenta il valore di T_1.
Per quanto riguarda T_2 possiamo scegliere un valore costante, oppure mantenere costante il periodo totale $T_1 + T_2$, calcolando T_2 a partire da T_1. Questa seconda soluzione permette di variare la potenza nel carico da 0 al 100%. Il massimo valore di T_1 è FF_H, quindi:

$$T_2 = FF_H - T_1$$

Il parametro di parzializzazione (cioè il valore di T_1) può essere dato una volta per tutte prima del lancio del programma; in questo caso la sequenza di operazioni è quella di Figura 4. Diversamente possiamo scegliere una soluzione maggiormente interrattiva, che permetta di modificare il parametro T_1 anche mentre il programma è in esecuzione. Possiamo per questo usare il blocco SCAN del monitor, che trasferisce le cifre via via introdotte dalla tastiera nella cella BUFF. (Il diagramma di flusso in questo caso è riportato in Figura 5).
Il listato completo del programma, assemblato per il Picocomputer, è LIST 1.

Esempio 2: misura di periodo e conversione A/D

Questo esempio è il "duale" del precedente; obbiettivo è infatti convertire un tempo in un valore numerico. Possiamo così misurare direttamente il periodo di un segnale, oppure, per via indiretta, tensioni, correnti, resistenze. In questi casi l'interfaccia esterna sarà un convertitore tensione (o corrente)/frequenza, realizzato ad esempio secondo le tecniche descritte in *La progettazione dei circuiti Amplificatori Operazionali,* pag. 9-22 (Ed. Jackson).
Per questo particolare esempio faremo riferimento ad un semplicissimo convertitore resistenza/periodo (v. Figura 6). Questo circuito permette di realizzare, con gli opportuni trasduttori, misuratori di luce, temperatura, e così via. Grazie alla presenza del microprocessore, eventuali linearizzazioni possono essere eseguite da programma, usando adatti algoritmi o tabelle. Dato che la precisione ottenibile con queste tecniche di conversione è limitata dalla parte analogica, limitiamo la misura ad una risoluzione di 8 bit.
Il "programma di acquisizione" deve pertanto misurare l'intervallo di tempo T (v. Figura 6), compreso tra due transizioni omologhe (0-1 oppure 1-0) del segnale U riportato su una linea di ingresso. Dal momento che sono sufficienti 8 bit, useremo un registro come contatore, incrementato periodicamente durante l'intervallo T. La costante di tempo RC va scelta in modo da non superare 255 cicli (FF_H) per il massimo valore di R. Valori adatti sono $R_{MAX} = 500\ K\Omega$, $C = 10\ \mu F$. Le operazioni da compiere sono indicate nella Figura 7.
Dobbiamo anche visualizzare il contenuto del regi-

Figura 6 - Conversione Resistenza/Periodo.

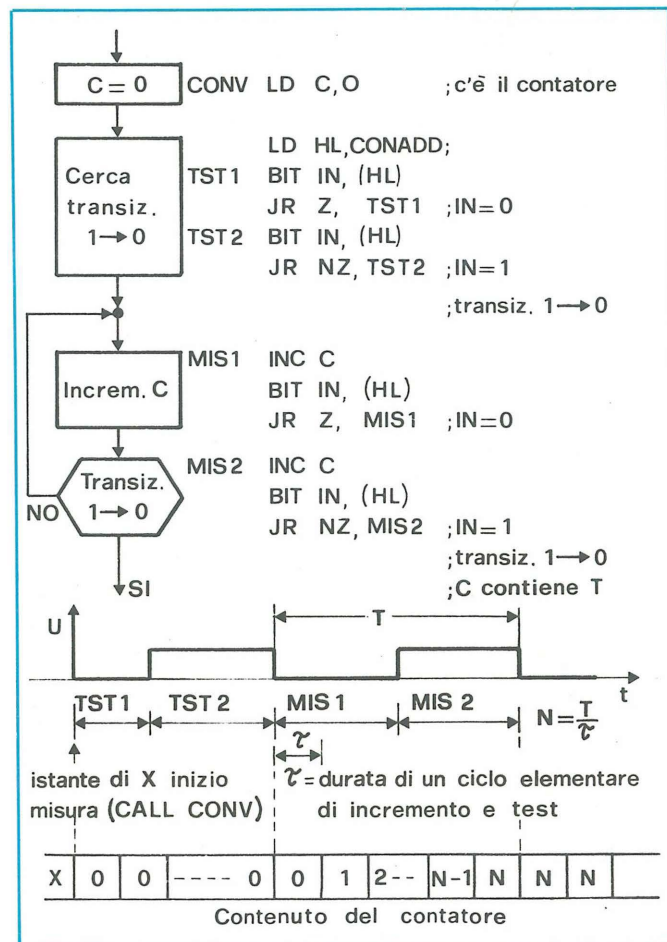
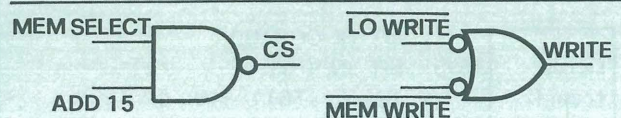
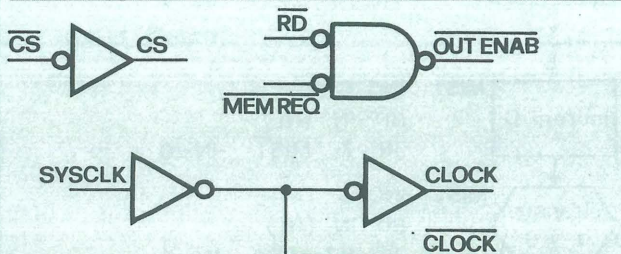
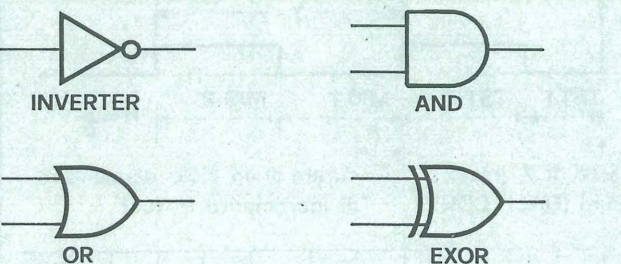



Figura 7 - Misura del periodo di un segnale U mediante conteggio di cicli elementari. "N" è il risultato della conversione A/D.

Alcune convenzioni grafiche

Comprendere schemi elettrici, diagrammi temporali di circuiti logici, in particolar modo di sistemi a microprocessori, richiede una precisa definizione dei simboli e delle notazioni usate.
Innanzi tutto è necessario fare una chiara distinzione tra lo schema funzionale e le sue possibili realizzazioni.
Gli schemi funzionali descrivono la funzione da implementare, indipendentemente dal dispositivo specifico che sarà chiamato a realizzarla.
Gli schemi relativi all'implementazione, gli usuali schemi elettrici, sono invece legati alle scelte tecnologiche ed ai vari componenti, e descriveranno in dettaglio la particolare realizzazione; anch'essi comunque devono seguire nel modo più stretto possibile gli schemi funzionali, affinché risultino agevolmente comprensibili.
Occorre quindi fare una certa attenzione nell'usare i simboli logici negli schemi elettrici; essi infatti devono rispecchiare la funzione logica che si intende realizzare e non quella per cui sono comunemente posti sui cataloghi.
Un circuito NAND, per esempio, può essere indifferentemente usato sia per realizzare una funzione logica NAND che una funzione logica OR di segnali in logica negativa all'ingresso (v. Figura 1).
È molto importante ai fini di una facile lettura dello schema elettrico mantenere sempre il simbolo grafico della funzione che si intende realizzare ed inserire il segno di negazione (il pallino) dove il segnale è invertito (v. Figura 2).
Per risalire poi dal simbolo funzionale al dispositivo fisico, basta ricordare alcune identità, dalle quali si deduce per esempio che i circuti di Figura 3 sono assolutamente equivalenti.
L'uso di convenzioni grafiche semplifica ulteriormente la stesura e la lettura dei disegni.

Le convenzioni grafiche per le porte sono ormai note ed indiscusse (v. Figura 4), mentre i dispositivi più complessi sono sempre indicati ricorrendo a blocchetti rettangolari a cui si fanno terminare le varie linee, con l'eventuale segno di negazione, scrivendo dentro il riquadro il loro significato (v. Figura 5).
Oltre a tutto quanto detto sinora, cose ormai ovvie, occorre dare adeguati simboli grafici alle porte per distinguere quale tipo di queste si intende usare.
Le porte disegnate come descritto sopra (non contrassegnate da alcun segno) devono intendersi normali.
Un punto vicino all'uscita significa che la porta è di tipo open-collector.
La resistenza di pull-up usata normalmente con i dispositivi open-collector può essere disegnata esplicitamente oppure indicata per brevità con un asterisco (v. Figura 6).
Tre brevi lineette vicino all'uscita indicano invece che la porta ha l'uscita 3-stati, il filo di abilitazione del 3-stati entra lateralmente alla porta, mentre nei dispositivi più complessi è distinto dagli altri segnali da un piccolo semicerchio (v. Figura 7).
Un tratto parallelo dalla parte degli ingressi invece indica che la porta è di tipo schmitt-trigger (v. Figura 8).
Un triangolino sull'ingresso significa che quello è l'ingresso del clock del dispositivo stesso.
Usualmente si intende che il dispositivo è attivato durante il fronte di salita del segnale, mentre se è preceduto dalla negazione, dal fronte di discesa (v. Figura 9).
Queste ultime convenzioni (ed altre ancora, non citate perché di uso rarissimo) hanno raggiunto un considerevole seguito, ma non l'unanimità. Esse inoltre non sono in perfetto accordo con la normativa internazionale, ma ricalcano più che altro quello che può essere definito l'uso comune e la pratica di alcuni fabbricanti di tali dispositivi, in particolare della AMD e della INTEL. ■

MEM SELECT — ADD 15 — CS; LO WRITE — MEM WRITE — WRITE

Fig. 1

CS — CS; RD — MEM REQ — OUT ENAB; SYSCLK — CLOCK — CLOCK

Fig. 2

INVERTER — AND — OR — EXOR

Fig. 4

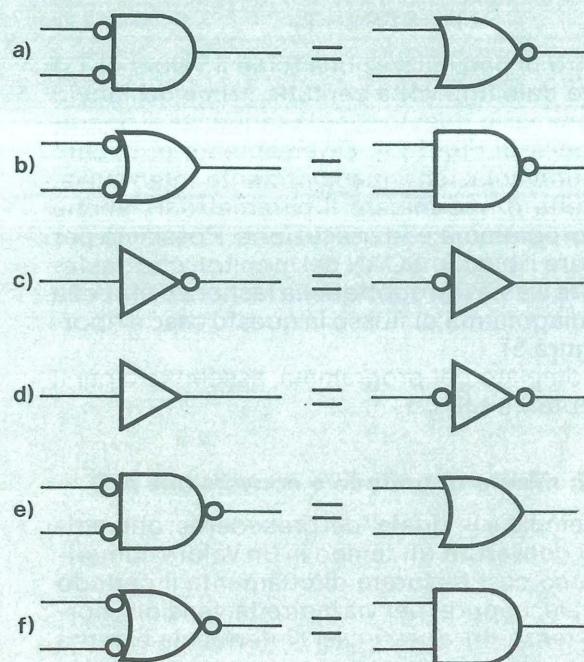



Fig. 3

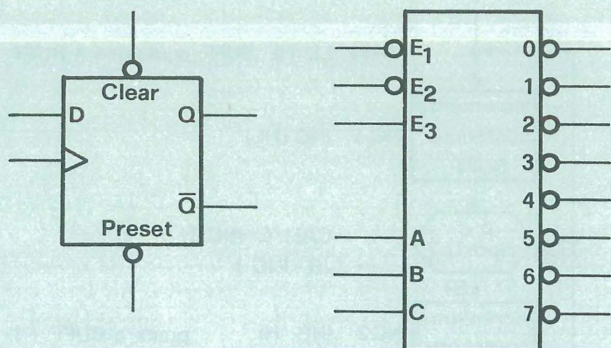



Fig. 5

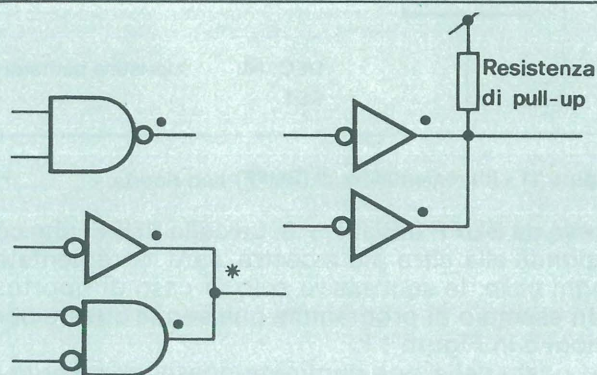



Fig. 6

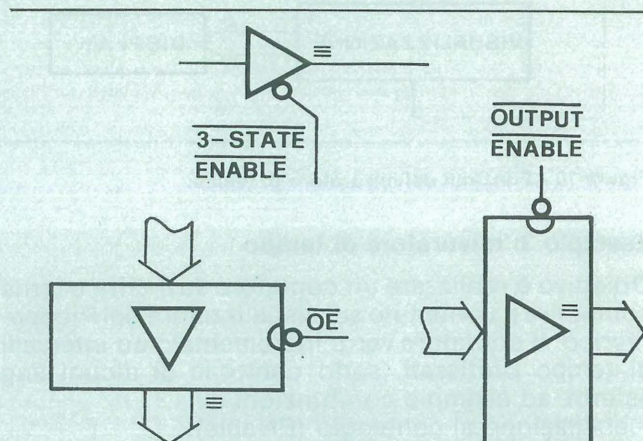



Fig. 7

Fig. 8

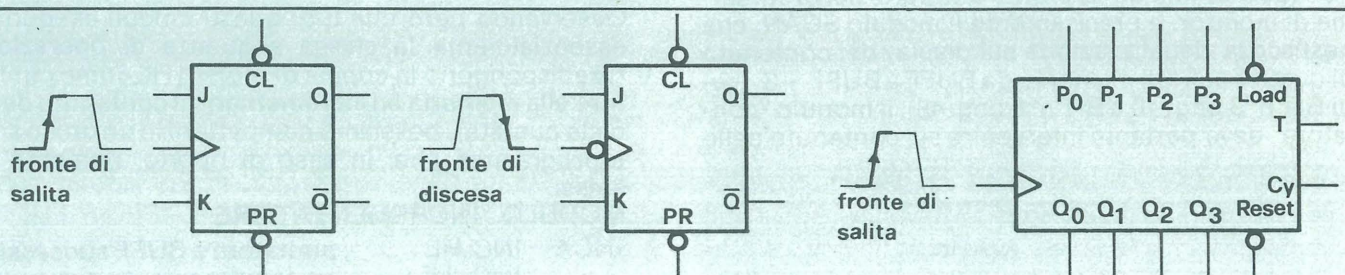



Fig. 9

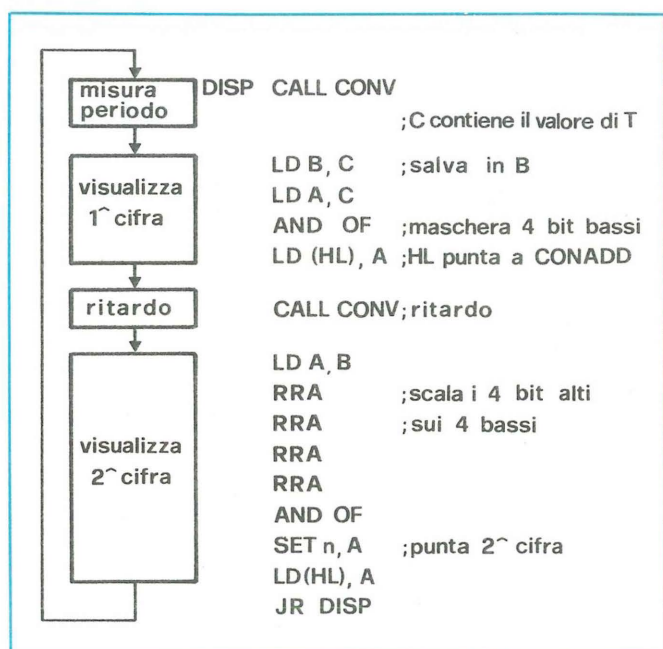



Figura 8 - Misura di periodo e visualizzazione.

stro contatore; trattandosi di due sole cifre esadecimali, non conviene usare il sottoprogramma SCAN del monitor, che scandisce tutte le 8 cifre del display. Prepareremo pertanto una routine apposita, che diventa il programma principale per questa applicazione e chiama il modulo "Misura periodo" per aggiornare il registro contatore.

Nel programma di Figura 8 il modulo di conversione viene chiamato due volte, ma si visualizza solo il risultato della prima conversione. La seconda chiamata ha il solo scopo di mantenere accese le due cifre per uno stesso tempo, perché abbiano la stessa luminosità.

Il formato della parola da inviare al Picoperiferico per la gestione del display è in Figura 9; per ulteriori dettagli vedi BIT n° 3.

Il listato completo per questa applicazione è LIST 2.

Figura 9 - Formato per la presentazione di un carattere sul display.

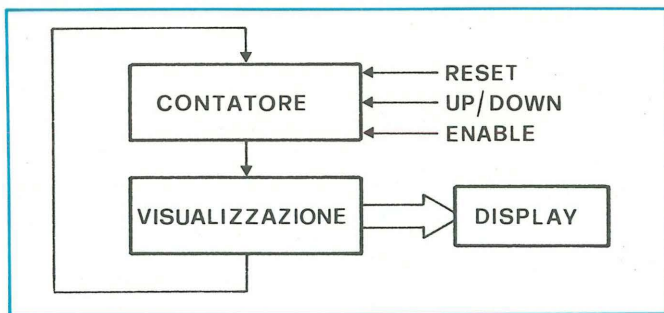



Figura 10 - Funzioni del misuratore di tempo.

Esempio 3: misuratore di tempo

Obiettivo è realizzare un contatore su 8 cifre e farne comparire il contenuto sul visualizzatore del Picoperiferico. Il contatore verrà incrementato ad intervalli di tempo prefissati, sotto controllo di alcuni flag esterni, ad esempio con funzioni di:

- abilitazione al conteggio (ENable);
- azzeramento (ReSet);
- selezione incremento/decremento (Up/Down).

Il contatore può essere configurato per presentare ore, minuti primi e secondi, decimi e centesimi.
Le operazioni da eseguire sono raggruppabili in due moduli di base: il contatore vero e proprio e la routine di visualizzazione (v. Figura 10).
Per quest'ultima utilizzeremo una parte del programma di monitor, e precisamente il modulo SCAN, che gestisce la visualizzazione sul display del contenuto di quattro celle di memoria, da BUFF a BUFF + 3. (vedi BIT n° 3 pag. 40 e BIT n° 5 pag. 48). Il modulo "contatore" deve pertanto intervenire sul contenuto delle

```
                     1  ;
                     2  ;
                     3  ;           CONVERTITORE A/D
                     4  ;
                     5  ;
                     6  ;           PARAMETRI
                     7  ;
                     8  USP         EQU       0BD0H
                     9  CONADD      EQU       0C00H
                    10  IN          EQU       4            ;PIN 5 CONN. B
                    11  ;
                    12  ;
                    13  ;
0800                14              ORG       800H
                    15  ;
                    16  ;
0800   31D00B       17  DISP        LD SP,    USP
0803   CD1A08       18  DISP1       CALL      CONV
0806   41           19              LD B,     C
0807   79           20              LD A,     C
0808   E60F         21              AND 0FH
080A   77           22              LD (HL),A
080B   CD1A08       23              CALL      CONV
080E   78           24              LD A,     B
080F   1F           25              RRA
0810   1F           26              RRA
0811   1F           27              RRA
0812   1F           28              RRA
0813   E60F         29              AND 0FH
0815   CBE7         30              SET 4,    A
0817   77           31              LD (HL),A
0818   18E9         32              JR        DISP1
                    33  ;
                    34  ;
                    35  ;
                    36  ;           MISURA DEL PERIODO
                    37  ;
081A   0E00         38  CONV        LD C,     0
081C   21000C       39              LD HL,    CONADD
081F   CB66         40  TST1        BIT IN,   (HL)
0821   28FC         41              JR Z,     TST1
0823   CB66         42  TST2        BIT IN,   (HL)
0825   20FC         43              JR NZ,    TST2
0827   0C           44  MIS1        INC       C
0828   CB66         45              BIT IN,   (HL)
082A   28FB         46              JR Z      MIS1
082C   0C           47  MIS2        INC C
082D   CB66         48              BIT IN,   (HL)
082F   20FB         49              JR NZ,    MIS2
0831   C9           50              RET
                    51  ;
                    52  ;
                    53  ;
```

LIST 2

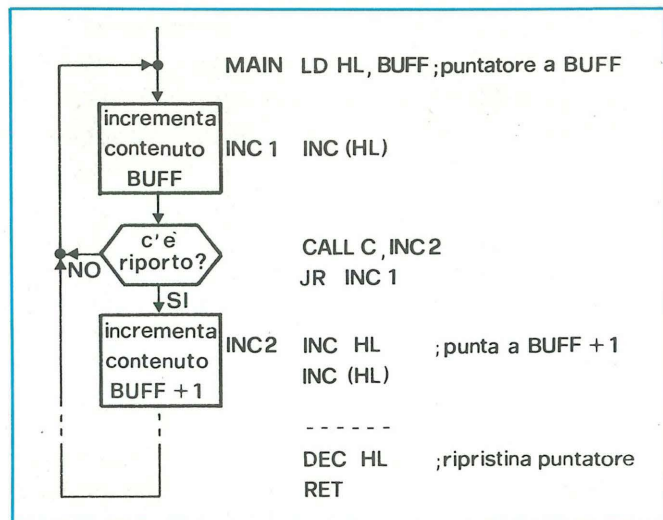



Figura 11 - Incrementatore di (BUFF) con riporto.

celle da BUFF a BUFF + 3. La cella BUFF, che corrisponde alla cifra più a destra, sarà incrementata ad ogni ciclo; le successive solo in caso di riporto.
Un esempio di programma che segue queste operazioni è in Figura 11.
Al posto della riga puntinata possiamo inserire una chiamata condizionata alla routine che incrementa BUFF + 2 in caso di riporto su BUFF + 1, e così via.
Osservando però che tutti questi moduli eseguono essenzialmente la stessa sequenza di operazioni (predispongono la coppia di registri HL come puntatore alla memoria ed incrementano il contenuto della cella puntata), possiamo compattarli in un unico sottoprogramma che, in caso di riporto, richiama se stesso.

MODULO "INCREMENTATORE"

```
INCA   INC HL       ; puntatore a BUFF successivo
       INC (HL)     ; ne incrementa il contenuto
       CALL C INCA; se c'è riporto ripete l'operazione
       DEC HL       ; ripristina il puntatore originale
       RET
```

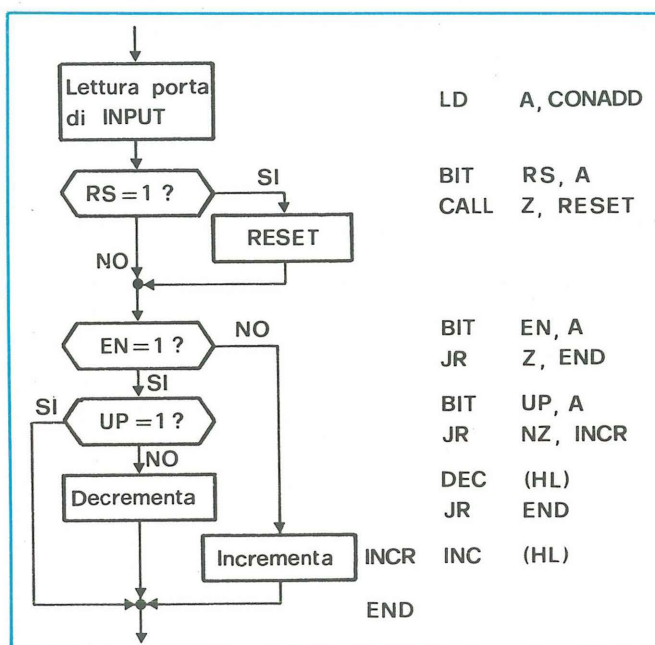



Figura 12 - Comandi al contatore.

Queste stesse operazioni vengono scorporate anche dal programma principale, che, inserendo anche la chiamata della routine di visualizzazione, diventa:

```
MAIN    LD HL,BUFF – 1  ; predispone il puntatore
        CALL INCA       ; incrementatore
        CALL SCAN       ; visualizzazione
        JR MAIN
```

Questo modulo, insieme all'incrementatore, implementa già un contatore esadecimale su 8 cifre, visualizzato sul display del Picoperiferico.

Per soddisfare le specifiche iniziali occorre ancora testare i segnali di ingresso per decidere se eseguire o meno un azzeramento, se abilitare il conteggio, se incrementare o decrementare. Per questo è sufficiente inserire nel modulo "incrementatore" una serie di operazioni condizionate dal test sul corrispondente comando (dato attraverso linee di ingresso). Il nuovo diagramma di flusso è in Figura 12.

Il contatore a questo punto conta, e può essere bloccato o resettato.
Ciascuna cella di BUFF assume i valori da 0 a 255, visualizzati sul Picoperiferico con due cifre esadecimali. Per ottenere un contatore decimale o secondo altri moduli (per esempio modulo 60 per un orologio con minuti primi e secondi), occorre inserire altri blocchi, rispettivamente di conversione binario-decimale e di limite all'incremento di una cella. Per la conversione binario-decimale è comodo usare l'istruzione Decimal Adjust Accumulator (DAA; per i dettagli sulle operazioni eseguite vedi manuale Z 80 o 8080), che però opera solo sull'accumulatore e deve essere preceduta da una istruzione ADD per operare correttamente. All'istruzione INC (HL) sostituiremo quindi:

```
LD A, (HL)      ; trasferisce nell'accumulatore
ADD 01          ; incrementa
DAA             ; conversione binario-decimale
LD (HL), A      ; ritrasferisce in BUFF
```

In coda a questo blocco possono essere inserite le istruzioni per il conteggio su modulo MOD diverso da 100:

```
        CP MOD          ; conteggio massimo?
        CALL Z OVERMOD  ; sì, dà riporto
        ....            ; no, continua
OVERMODLD (HL), 00      ; azzera
        CALL INC        ; riporto su cifra
                          successiva
        RET
```

Perché il misuratore di tempo diventi completamente operativo occorre ancora tararlo, cioè definire l'intervallo di tempo tra un incremento e l'altro. Questo tempo può essere calcolato dal programma completo, determinando il numero di cicli impiegati per eseguire un loop del programma principale. Volendo invece un contatore direttamente tarato, ad esempio in centesimi di secondo (e questa è la nostra specifica), è sufficiente inserire una routine di ritardo con parametro opportuno.
Va messo in evidenza che la precisione ottenibile è limitata, perché a causa delle istruzioni di salto condizionato il numero di cicli richiesti dal programma non è un dato costante. Ad esempio, in caso di riporto su una cella del contatore si deve incrementare la successiva, e questo comporta l'esecuzione di un maggior numero di istruzioni. È possibile eliminare questo errore introducendo in posizioni opportune delle istruzioni che hanno il solo scopo di eguagliare i tempi di esecuzione per qualsiasi situazione. L'errore introdotto diventa comunque trascurabile se l'intervallo di incremento è determinato in massima par-

```
                    1  ;
                    2  ;
0800                3           ORG      800H
                    4  ;
                    5  ;
                    6  ;        DEFINIZIONE PARAMETRI
                    7  ;
                    8  BUFF     EQU      0BF0H
                    9  CONADD   EQU      0C00H
                   10  USP      EQU      0BD0H
                   11  DEL_T    EQU      368H
                   12  COP      EQU      44H
                   13  RS       EQU      3          ;PIN 4 CONN. B
                   14  EN       EQU      4          ;PIN 5 CONN. B
                   15  ;
                   16  ;
                   17  ;        PROGRAMMA PRINCIPALE
                   18  ;
0800   31D00B      19  INCM     LD SP,   USP
0803   21EF0B      20  TEST     LD HL,   BUFF-1
0806   3A000C      21           LD A,    (CONADD)
0809   CB5F        22           BIT RS,A
080B   CC3B08      23           CALL Z   RESET
080E   CB67        24           BIT EN,  A
0810   2806        25           JR Z     ENDTEST
0812   CD1D08      26           CALL     INCA
0815   CD4A08      27           CALL     DELAY
0818   CD4400      28  ENDTEST  CALL     COP
081B   18E6        29           JR       TEST
                   30  ;
                   31  ;
                   32  ;
                   33  ;        INCREMENTATORE PER CENTESIMI E DECIMI
                   34  ;
081D   23          35  INCA     INC      HL
081E   7E          36           LD A,    (HL)
081F   C601        37           ADD A,   01
0821   27          38           DAA
0822   77          39           LD (HL),A
0823   DC2808      40           CALL C   INC_B
0826   2B          41           DEC HL
0827   C9          42           RET
                   43  ;
                   44  ;
                   45  ;
                   46  ;        INCREMENTATORE PER SECONDI, PRIMI E ORE
                   47  ;
0828   23          48  INC_B    INC      HL
0829   7E          49           LD A,    (HL)
082A   C601        50           ADD A,   01
082C   27          51           DAA
082D   77          52           LD (HL),A
082E   FE60        53           CP       60H
0830   CC3508      54           CALL Z   INC_C
0833   2B          55           DEC HL
0834   C9          56           RET
                   57  ;
                   58  ;

                   59  ;
0835   3600        60  INC_C    LD (HL),0
0837   CD2808      61           CALL     INC_B
083A   C9          62           RET
                   63  ;
                   64  ;
                   65  ;
                   66  ;        AZZERAMENTO DEL CONTATORE
                   67  ;
083B   E5          68  RESET    PUSH HL
083C   21F00B      69           LD HL,   BUFF
083F   3E00        70           LD A,    00
0841   77          71           LD (HL),A
0842   23          72           INC      HL
0843   77          73           LD (HL),A
0844   23          74           INC      HL
0845   77          75           LD (HL),A
0846   23          76           INC      HL
0847   77          77           LD (HL),A
0848   E1          78           POP      HL
0849   C9          79           RET
                   80  ;
                   81  ;
                   82  ;
                   83  ;        RITARDO PER TEMPORIZZAZIONE
                   84  ;
084A   C5          85  DELAY    PUSH     BC
084B   016803      86           LD BC,   DEL_T
084E   0B          87  DELOP    DEC      BC
084F   78          88           LD A,    B
0850   B1          89           OR       C
0851   20FB        90           JR NZ    DELOP
0853   C1          91           POP      BC
0854   C9          92           RET
                   93  ;
                   94  ;
```

LIST 3

te dalla routine di ritardo, che viene sempre eseguita allo stesso modo.
Il listato completo del programma che implementa un contatore tipo "cronometro", con reset ed abilitazione è ripotato come LIST 3.
Si può notare come siano presenti due moduli incrementatori distinti, rispettivamente per frazioni di secondo (modulo 100) e per ore, primi e secondi (modulo 60). Durante l'esecuzione del programma si noterà anche che le cifre non sono uniformemente illuminate; questo perché durante il ritardo (cioè per la maggior parte del tempo), viene visualizzata sempre la stessa cifra (ultima a sinistra), che diventa quindi più luminosa.

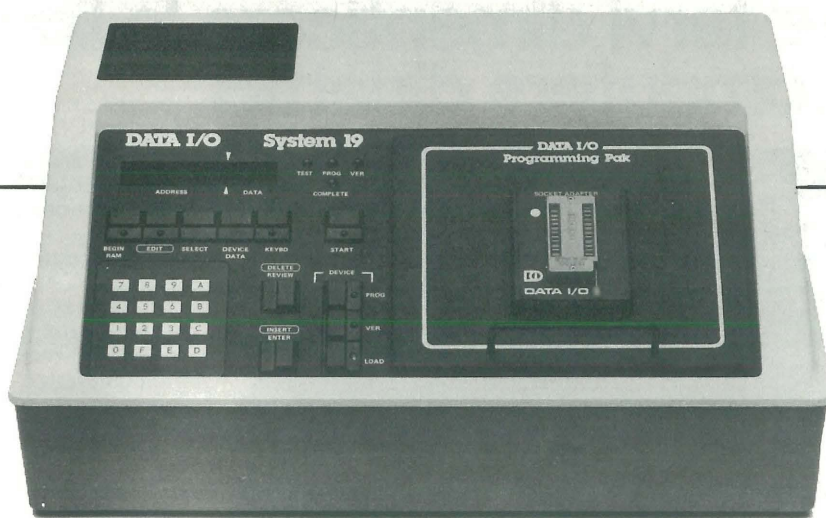
Proposte e suggerimenti

Attorno alle applicazioni base qui descritte è possibile costruire molte varianti, interessanti dal punto di vista didattico ed applicativo.
L'esempio 1 è facilmente modificabile in modo da far variare il periodo dell'onda quadra anziché il duty-cycle; colleghiamo un altoparlante come carico e si apre il campo della sintesi di suoni (almeno per chi non ha problemi di fedeltà). Mantenendo la tecnica della parzializzazione è anche possibile sintetizzare forme d'onda qualsiasi, variando con continuità il duty-cycle secondo leggi opportune.
L'esempio 2 può diventare uno strumento digitale vero e proprio: capacimetro, voltmetro a rampa, misuratore di frequenza con diverse portate ed addirittura con auto-range, e così di seguito.
L'esempio 3 si presta ad un notevole numero di estensioni; dal semplice timer ad azionamenti secondo sequenze temporali prefissate, fino ad un orologio che si sincronizzi automaticametne, decodificando il prr..t trasmesso prima dei segnali orari RAI (che contiene, codificate in FSK, le informazioni di ora e data). ■

Nell'ambito della giornata di studio organizzata da ELETTRONICA OGGI sulle applicazioni dei microprocessori

IL LETTORE PROTAGONISTA

Al fine di promuovere il più ampio dibattito sulle applicazioni della microelettronica sia nel settore industriale che in quello della vita quotidiana, Elettronica Oggi invita tutti i lettori a presentare applicazioni e progetti da loro elaborati. I migliori lavori saranno presentati dagli stessi autori nel corso della giornata di studio organizzata dal Gruppo di Studio sulle applicazioni dei microprocessori di Elettronica Oggi durante la mostra Bias '80 Microelettronica. Tale giornata sarà dedicata allo studio delle modifiche ambientali prodotte dai microelaboratori.

"BEEP! BEEP!"

Il programmatore universale di Prom DATA I/O risponde per farti risparmiare tempo e denaro.

È un programmatore che ti dice che tutto va bene o, se qualche cosa non va, ti specifica perchè.

Accendilo ed il Mod. 19 prova tutti i suoi componenti: porte input/output, microprocessore, RAM, Bus, pannello frontale e persino il suo software interno.

Durante la sequenza di programmazione il sistema 19 esegue il check dell'"illegal bit", il "blank check", il "sum check" del componente appena programmato e la verifica a tensione alta, bassa, media come richiesto per essere sicuri che il componente soddisfi le specifiche operative del costruttore.

Il sistema 19 avvisa in modo discreto l'operatore quando è stato fatto un semplice errore di procedura e si blocca prima che l'operatore possa fare un errore più serio e possa danneggiare il componente.

Se qualche cosa va male durante le operazioni, il sistema 19 lampeggia indicando sul display uno dei 27 codici di errore in modo da guidarti velocemente alla soluzione del problema specifico.

Con il sistema 19 potrai programmare ogni tipo di componente logico programmabile comprese le FPLA. Con il suo modulo multiplo potrai programmare fino ad otto componenti MOS alla volta.

La cosa migliore è che il DATA I/O Sistema 19 ha un prezzo abbordabile da tutti.

È impossibile fare una ricerca di mercato sui programmtori senza considerare il mod. 19.

Fa in modo che possiamo spiegarti le differenze, telefona o scrivi alla SISTREL.

Buona Idea!

DATA I/O

Programming systems for tomorrow..... .....today.

SISTREL
SOCIETÀ ITALIANA STRUMENTI ELETTRONICI SPA

Via Pelizza da Volpedo 59 - 20092 CINISELLO B.
tel. (02) 6181893 - Telex 320346
Via Giuseppe Armellini 39, 00143 ROMA. Tel. (06) 5915553 - Telex 680356
Via Cintia Parco S. Paolo 35, 80126 NAPOLI - Tel. (081) 7679700

☐ Ricevere un'offerta ☐ La visita di un Vs. Tecnico ☐ Essere inseriti nel Vs. mailing list.

NOME .... COGNOME ....
VIA .... TEL. ....
CAP .... CITTÀ ....
DITTA ....
REPARTO ....

System 19

Corso sul PASCAL
Struttura generale del linguaggio PASCAL

di **F. Waldner** - *Università di Bari, Istituto di Fisica*

Parte II

Struttura di un programma

Cominciamo con una serie di definizioni e di terminologie.
Un *programma* (program), detto anche *algoritmo* (algorithm)[1] si presenta in PASCAL, come in qualunque linguaggio, come una serie di *azioni* (actions) che operano su dei *dati* (data). I dati vengono descritti in PASCAL da *dichiarazioni* (declarations) e da *definizioni* (definitions). Le azioni sono invece descritte da *istruzioni* (statements). Fin qui nulla di nuovo.

La struttura generale di un programma è quella di un *blocco* (block) in cui sono contenute tutte le definizioni, dichiarazioni ed istruzioni ed eventualmente anche i sottoprogrammi. Tale blocco contiene all'inizio un'*intestazione* (heading) con una forma prefissata. In generale vedrete che dichiarazioni e definizioni sono soggette a regole molto più rigorose che nel FORTRAN.

Ecco il BNF per l'intestazione:

<programma> :: = <intestazione> <blocco> (2.1)
<intestazione> :: =PROGRAM <identificatore>
(<identificatore di file>,
{<identificatore di file>{)};
<identificatore di file>:: = <identificatore>

Cos'è un "identificatore" lo vedremo più avanti. Per ora, supponendo che X1, X2, X3, X4 siano gli identificatori, penso sarete rapidissimi nel decidere quali delle seguenti intestazioni siano corrette, quali no, e perché. La risposta a fine articolo. Incidentalmente avremo altri "CONTROLLI"; si tratterà sempre di decidere cos'è giusto, cos'è sbagliato, e perché. Quindi d'ora in avanti non lo ripeterò. Eccovi allora il primo esercizio.

CONTROLLO 1

I)	PROGRAM X1	VI)	PROGRAM X1 (Y2);
II)	X1 (X2);	VII)	PROGRAM X1
III)	PROGRAM X1 (X2)	VIII)	MAIN X1 (X2, X3);
IV)	PROGRAM X1	IX)	PROGRAM X1, X2;
V)	PROGRAM X1 (X2, X3, X4);	X)	PROGRAM X1, X2;

Vi sottolineo tre punti:
- Il PROGRAM in PASCAL è il programma complessivo, insieme di quelli principali e di quelli dipendenti. *Non* è quindi il solo MAIN (detto per i fortranisti).
- Il fatto di dover dichiarare *tutte* le variabili, costanti etc, potrà sembrarvi noioso e inefficiente. Non è così: tenete presente che le regole generali di PASCAL sono state formulate dopo esperienze lunghe e sofferte.

In generale hanno lo scopo di costringere la gente ad essere metodica ed ordinata (anche a costo di qualche sacrificio) e di spingerla naturalmente verso il metodo *top-down* ed alla programmazione strutturata.
- Almeno un identificatore di file *deve* essere presente nell'intestazione. Questo identificatore di file verrà usato per l'output del programma.

Un po' di miscellanea

Anzitutto vediamo il vocabolario di PASCAL. Vi dirò subito che questo dipende dalle implementazioni del sistema e del calcolatore che usate.

Comunque lo standard è questo: (2.2)

<digit>:: = 0 | 1 | 2 | 3 | 4 | 5 | 6
7 | 8 | 9 |
<lettere>:: = A | B | C | D | E | F | G
H | I | J | K | L | M | N
O | P | Q | R | S | T | U
V | W | X | Y | Z |
<simbolo>:: = + | — | * | / | := | . | ,
; | : | ' | = | <> | < | <=
>= | > | (|) | [|] | {
} | ↑ | " |
<metacostanti>:: = AND | ARRAY | BEGIN | CASE |
CO | DIV | DO | DOWNTO |
ELSE | END | FILE | FOR |
FUNCTION | GOTO | IF | IN |
LABEL | MOD | NIL | OF | OR |
PACKED | PROCEDURE |
PROGRAM | RECORD |
REPEAT | SET | THEN | TO |
TYPE | UNTIL | VAR | WHILE |
WITH

Non ho sottolineato nessuna metacostante, ritenendolo implicito. Vi prego di notare tre cose:
- E' assolutamente vietato usare per altri scopi che non siano quelli previsti da PASCAL le metacostanti. Per esempio non si può chiamare una variabile SET o un sottoprogramma WITH.
- Notate che nella GOTO *non c'è* un blank in mezzo! Attenti i fortranisti: scrivere GO TO è una *comune causa di errore.*
- Vi avevo detto che in genere i simboli del metalinguaggio sono scelti in modo da *non* essere presenti nel linguaggio. Al lettore non sarà però sfuggita un'eccezione: le parentesi graffe { } che sono presenti sia nel BNF come simbolo di ripetizione, e anche nel PASCAL. Come mai? Perché non abbiamo a disposizione un numero infinito di simboli, e allora dobbiamo scegliere il male minore.

Fuori di metafora, l'unico caso in cui si usano le parentesi graffe in PASCAL è il commento.
Si scrive per esempio

{ QUESTO E' UN COMMENTO } (2.3)

e questo può essere inserito *dovunque* nel program-

[1] *Vi sarete magari chiesti qualche volta da cosa derivi questa parola: deriva dal soprannome di un matematico arabo (Muhammad Ibu Musà), vissuto a Baghdad fra il 775 e l'850 circa, datto Al - Khuwaritzmi, perché era originario del Korassan.*
Autore dell'algebra moderna, e divulgatore del sistema posizionale e decimale dei numeri (introdotto dall'India), una sua opera fondamentale venne tradotta (malamente) in latino, e cominciò con le parole "Algoritmi dicit ..." che vorrebbe dire "Dice quello del Korassan.." ed invece passò (nell'abissale ignoranza di quei secoli) a significare "Algoritmo dice ...". Da qui il nostro algoritmo e anche - per ulteriore corruzione - logaritmo, il primo passato a significare "un qualunque attrezzo matematico".

ma, anche - al limite - in mezzo ad uno statement. Anche questo costituisce una comodità notevole; si tratta di farci un po' l'abitudine: pagato però lo scotto iniziale si arriva rapidamente a scrivere dei listings che non necessitano di ulteriore documentazione. E scusate se è poco!
Tutto ciò ci porta al concetto di *separatori* (separators): questi sono gli *spazi* (blanks), i caratteri di *fine linea* (*E*nd-*O*f-*L*ine, oppure per brevità EOL), ed i *commenti* (comments).
I separatori servono (ovviamente!) a separare dei simboli, siano questi dei nomi di variabili o delle parole riservate (metacostanti) come quelle della (2.2). Se ne possono mettere quanti se ne vuole, analogamente a quanto avviene in FORTRAN, nel quale si possono inserire blanks a piacimento, ma *non* all'interno di una parola riservata: non si può scrivere infatti qualcosa come FUNC TION! Diversamente dal FORTRAN però ci sono due separatori in più:

- *Il carattere di fine linea:* ciò vuol dire che non vi dovete *mai* preoccupare di andare a capo, sia cambiando scheda sia cambiando riga in una TTY. Non esiste insomma più il problema delle "schede di continuazione" del FORTRAN.
- *Il commento:* la possibilità di inserirlo dove serve e al momento giusto, magari per spiegare una espressione matematica particolarmente manipolata e semplificata, o in mezzo alle costanti per spiegare che cosa sono ed a cosa servono.

Non ve l'ho ancora detto, ma penso che vi sia già chiaro che il PASCAL è un linguaggio a formato libero: se -per esempio - usate schede, non siete obbligati a lasciare le prime 5 colonne alla numerazione, la sesta alla continuazione, etc. E' rimasto in PASCAL solo l'uso delle colonne dalla 72 alla 80 per la numerazione eventuale di schede o righe. Per il resto gli statements possono essere messi comunque nella scheda o nella riga. E' evidente che occorre quindi separarli in qualche modo: a ciò provvede il punto e virgola (;).

Come ultimo punto parliamo dell'*identificatore* (identifier) questo è il nome che viene dato a qualcosa, sia questo qualcosa una variabile, una matrice, un sottoprogramma o ciò che volete. La struttura dell'identificatore è unica ed il BNF è

<identificatore>:: = <lettera>
{<lettera> | <digit>}$_0^7$

e per voi sarà uno scherzo superare il

CONTROLLO 2

I)	AB	VI)	CESARINO
II)	AB2XY	VII)	AND
III)	SET	VIII)	32 ABZ
IV)	SET A3	IX)	CESARINACCIO
V)	POINT. 1	X)	ABZ

In particolare: si possono usare gli identificatori VI) e IX) per indicare *due divese variabili?*

La struttura di un blocco

Quindi, scrivendo un programma in PASCAL, anzitutto occorre scrivere l'intestazione. E poi ovviamente tutto il resto. Nel quale trovano posto due parti fondamentali: nella prima noi specifichiamo tutto ciò che useremo (variabili, costanti ...) un po' come accade per la lista degli attori in una commedia; nella seconda operiamo sui dati: cioè si svolge effettivamente l'azione teatrale, la quale va scritta con una sintassi rigorosa:

<blocco>:: = <dichiarazioni di labels>
<definizioni di costanti >
<definizioni di tipi>
<dichiarazioni di variabili>
<dichiarazioni di procedure
e funzioni>
<istruzioni>

In questo articolo ci occuperemo *solo* delle dichiarazioni e definizioni. Comincio subito a farvi notare tre cose:

- Ancora una volta, a rischio di essere noioso, vi ricorderò che *tutto* (dico *tutto*) ciò che in qualche modo si usa *deve* essere in qualche modo dichiarato. Ancora un accenno al FORTRAN: non è permesso in PASCAL avere qualcosa di simile alle "dichiarazioni implicite" tipo "tutto ciò che comincia con I, J, K, L, M, N è intero". Queste false comodità sono fin dall'inizio bandite.

- Sarà bene che vi abituiate fin dall'inizio ad usare l'ordine alfabetico all'interno di dichiarazioni e definizioni. Renderete la vita più facile a voi e vi attirerete le benedizioni (sincere) di chi debba usare o modificare il vostro programma.

- L'ordine delle dichiarazioni e definizioni è *tassativo* e *non può* essere alterato.

Tutto ciò vi sembrerà molto teutonico e poco latino. Se poi verrete a sapere (a proposito, non ve l'ho ancora detto) che PASCAL è stato creato dal prof. Niklaus Wirth (non certo toscano) a Zurigo (Zürich!) penso vi rafforzerete in quest'idea. Bene, sempre a beneficio dei fortranisti: vi è mai capitato di avere dei messaggi di "missing routines", quando in realtà era successo che vi eravate *dimenticati* di dichiarare in una DIMENSION che TIZIO era una matrice, e quindi il compilatore pensava automaticamente che TIZIO (K, L) fosse una funzione di K e L, ed ovviamente non la trovava?

È molto importante quindi il:

CONTROLLO 3
1) occorre dichiarare prima i tipi o le funzioni?
2) occorre dichiarare prima le procedure o i labels?
3) occorre dichiarare prima le variabili o le costanti?
4) occorre dichiarare prima i tipi o le funzioni?
5) È possibile dichiarare i labels prima delle variabili?

Se trovate delle difficoltà riandate alla (2.5) ed imparatela a memoria.
Siccome però *non* ci riuscirete, non scoraggiatevi, e ricordatevi che PASCAL è
*L*a *C*osa *T*re *V*olte *P*iù *F*acile
proprio come le iniziali di
*L*abels *C*onstant *T*ypes *V*ariables *P*rocedures *F*unctions.
Se avete qualche frase meno sciocca, vi prego di divulgarla. Andiamo quindi per ordine e cominciamo con i dettagli della parte dichiarativa.

Labels

Sarà bene dire subito che in PASCAL questi sono dei veri fossili viventi da usarsi in casi estremi. In altri termini: le istruzioni possono essere numerate (proprio come nel FORTRAN) e il programma può saltare ad una certa istruzione tramite una GOTO. *Unico* caso in cui si usano i labels.
È bene che ricordiate fin dall'inizio che l'uso dei labels e GOTO è da sconsigliarsi decisamente. Meno li userete e più sarete progrediti verso una programmazione strutturata. Anzi non li dovreste usare *proprio mai*. Comunque vi può capitare di *doverlo* fare ed allora (visto che non ne possiamo fare a meno) eccovi qua i dettagli.

< label > :: = { < digit > }$_1^4$
< dichiarazione di label > :: = <u>LABEL</u> < label >
{,< label > }; (2.6)

Con un BNF così semplice non dovrebbe presentare difficoltà per voi il

CONTROLLO 4

I) 2	VII) LABEL 73
II) 37	VIII) LABEL 97, 32, 45
III) 9742	IX) LABEL 975;
IV) 10473	X) LABEL 9432, 57934;
V) 7A32	XI) LABEL 47, END2;
VI) A96	XII) LABEL 25, 43, 9000;

Fino al VI) di tratta di decidere se i labels sono legali o no. Dal VII) al XII) se sono legali o no le dichiarazioni.

Le costanti

È sempre buona pratica (anche se non è tassativo) mettere le costanti che si useranno nel programma in una definizione, in modo da permettere una facile lettura del programma stesso. Anzitutto però è necessario vedere cosa definiamo come costante. Queste possono essere di due tipi:
- *un numero*: questo è sempre scritto in notazione decimale e lo vedremo più sotto con il BNF.
- *una stringa* (string): questa è definita come una serie di caratteri *o simboli* racchiusa fra due apici. Se si vuole scrivere un apice, questo va scritto due volte. Ecco alcuni esempi di stringhe legali:

'STRINGA'
'ALTRA STRINGA'
'UN' ALTRA STRINGA'
'UNA STRINGA CON SPAZI   '
'ED UN' ALTRA CON SPAZI   E SIMBOLI:)+;'

Il BNF è un ottimo esercizio:

<definizione di costanti> :: = <u>CONST</u> (2.7)
<identificatore>
= <costante>
;{<identificatore>
= <costante>;}
<costante> :: = <numero non segnato>|
<segno> <numero non segnato>|
<identificatore di costante>|
<segno>
<identificatore di costante>|
<stringa>

<numero non segnato> :: = <intero non segnato>|
<reale non segnato>
<intero non segnato> :: = <sequenza di digit>
<sequenza di digit> :: = <digit> {<digit>}
<reale non segnato> :: = <intero non segnato>.
<sequenza di digit>|
<intero non segnato>.
<sequenza di digit> E
<fattore di scala>|
<intero non segnato> E
<fattore di scala>|
<numero non segnato> :: = <intero non segnato>|
<reale non segnato>
<fattore di scala> :: = {<digit>}$_1^2$|
<segno> {<digit>}$_1^2$
<segno> :: = + | −
<identificatore di costante> :: = <identificatore>
<stringa> :: = '<carattere>|<simbolo>
{<carattere>|<simbolo>}

Immagino che direte che è un po' complicato. Avete ragione. E quel ch'è peggio, ecco il CONTROLLO N°5 su come si può scrivere una costante:

CONTROLLO 5

I) .25	VIII) 3.25 E-75
II) PIGRECO	IX) .27 E+1
III) −PIGRECO	X) 0.25 E-23
IV) 72.25	XI) 9500000 E-101
V) 0.25	XII) 4 E-23
VI) −7ABZ	XIII) −22 E-23
VII) '−7ABZ'	XIV) −22. E−23
	XV) 22. 75 E45
	XVI) TONINO

E non basta. Occorre anche un controllo su come si scrive la dichiarazione delle costanti.
Quindi ecco il

CONTROLLO 6

I)	CONST	UNO=1; DUE=2; PIGRECO=3.1415;
		STRINGA = 'JACKSON ITALIANA'
II)	CONST	TRE = 3; CENTO = 1.0 E2
III)	CONST	AZ2 = 43 E-2; 2AZ4 = 'GIGETTO';
		RAGAZZA = 'MARIA LUISA';
IV)	CONST	UNO = 1; MENOUNO = −1;

Penso che abbiate fuso le bronzine. Vi faccio notare quindi alcune particolarità importanti per farvi rilassare:
- *Si possono* usare identificatori con segno.
- I commenti (debitamente racchiusi tra le parentesi graffe) possono essere inseriti dovunque. Anche in mezzo alla dichiarazione: essi contano come separatori, cioè come blanks in più. È inutile penso farvene notare la comodità: la documentazione del programma può essere drasticamente ridotta permettendo di costruire un listing che *si spiega da solo*.

Sarà anche bene notare che nella maggior parte delle implementazioni di PASCAL le parentesi graffe non ci sono. Ciò è dovuto al fatto che purtroppo non esiste ancora una standardizzazione dei codici dei caratteri. Normalmente si usa la combinazione dei caratteri [* e *] per racchiudere i commenti:

[* QUESTO È UN COMMENTO *]

Eccovi un esempio di dichiarazione di costanti commentata:

```
CONST E= 2.7181 [* BASE LOGARITMI NEPERIA-
        *];
      GIORNO = 'DOMENICA' [* GIORNO
      FESTIVO *];
      D = 6.278E3 [* RAGGIO TERRA IN KM *];
```

e penso che la vostra fantasia possa fare il resto.

Tipi e variabili

Qui cominciamo ad entrare nel vivo del linguaggio, affrontando - sia pure per gradi - le sue novità.
È un po' difficile spiegare in astratto che cosa sia un "tipo": possiamo dire, per spiegare il concetto, che si possono effettuare operazioni solo *fra oggetti dello stesso tipo*. Così, per esempio, mentre esistono regole per operare tra variabili logiche e tra numeri interi, non si possono mescolare tra loro questi due operando fra una variabile logica ed un numero intero.
Le *variabili* (variables) in PASCAL sono di due generi fondamentali: quelle *strutturate* (structured), quali matrici, records etc, che vedremo più avanti, e quelle *scalari* (scalars) o singole, di cui ci occuperemo in questo articolo. Le variabili scalari possono essere di qualunque tipo, e questa è una grossa novità in PASCAL: è lo stesso utente che può usare tipi nuovi adatti al suo problema specifico. Anche su questa faccenda però torneremo più avanti: per ora vi basti ricordare che potete *creare* tipi quali il mese dell'anno o il colore dei capelli, ed *operare* tra le variabili di questi tipi.
Occupiamoci dunque dei tipi standard che devono essere implementati in ogni sistema. Questi sono i soliti tipi che tutti conoscono: *intero* (integer), *reale* (real), *logico o booleano* (logic, boolean), *carattere* (character).
Vedremo le caratteristiche di questi tipi standard fra un momento. Per ora occorre ricordare che *ogni variabile* va dichiarata, assieme al suo tipo.

Ed ecco come:

<dichiarazione di variabile> :: = VAR <identificatore> {,<identificatore>}: <tipo>; {<identificatore> {,<identificatore>}: tipo;} (2.8)

<tipo> :: = INTEGER | REAL | BOOLEAN | CHAR

Naturalmente arriva - com'è ovvio - il

```
CONTROLLO 7
 I)  VAR X1, KZ, J: REAL; ALIVE, DEAD:
         BOOLEAN;
 II) VAR I, J, K, L: REAL; A1, B, C, D, : INTEGER;
III) VAR ZETA, KAPPA = CHAR; P1, Q2, 2A78:
         INTEGER;
 IV) VAR AB, BC, CB, BA: CHAR
         AC, BA, CZ, : INTEGER
         L1, LM, LN, ZC, CZ : BOOLEAN;
         A, B, C, D, L2 : REAL;
```

Il tipo intero

È costituito dai soliti numeri interi che tutti conoscono. Quando prenderete però un manuale in mano vi capiterà di leggere qualche cosa come "il subset dei numeri interi che dipende dall'implementazione del sistema". A questo punto resterete un po' perplessi. Niente paura: è un modo difficile di dire cose semplici. In pratica vuol dire solo che i numeri interi specificati da INTEGER non sono *tutti*, ma un sottoinsieme dei numeri interi: p. es., in un 16 bits gli interi vanno da − 32768 a + 32768 solamente: un intero come + 45000 *non è un* INTEGER. Tutto qui.
Le operazioni che sono permesse in questo tipo sono le seguenti:

*	moltiplicazione	(2.9)
+	addizione	
−	sottrazione	
DIV	divisione e troncamento	
MOD	che in linguaggio difficile è definito come A MOD B = A − ((A DIV B) * B) che in linguaggio facile è 25 MOD 10 = 5 73 MOD 7 = 3 È un'operazione quindi che dà *resto*.	

Alcune particolarità vanno notate a parte:
- L'operazione di divisione non si scrive / ma DIV. Questa differenza è voluta, in quanto la divisione fra due numeri interi è accompagnata dal troncamento; così

  7 DIV 2 dà 3   12 DIV 5 dà 2

- Non esiste un operatore esplicito per l'esponenziazione (come le * * del FORTRAN). Questa è una limitazione voluta, ma molto discussa: è probabile che nelle future implementazioni questa limitazione venga tolta, sull'onda della contestazione popolare.

Il tipo reale

Anche qui non c'è molto di nuovo da dire. Il tipo reale è costituito da tutti i numeri reali concessi dall'implementazione del sistema. Le operazioni permesse sono le quattro operazioni solite (*, /, +, −) ed anche qui non c'è un'operazione di esponenziazione. Alcune funzioni standard sono implementate in ogni sistema. Queste sono

ABS (X)	X	(2.10)
SQR (X)	X^2	
SIN (X)	sin x (x in radianti)	
COS (X)	cos X (x in radianti)	
ARCTAN (X)	atan x (risultato in radianti)	
LN (X)	ln x (logaritmo neperiano)	
EXP (X)	e^x	
SQRT (X)	$\sqrt{x}$	

Come vedete non c'è molto di nuovo. Una particolarità che va esplicitamente fatta notare è che nella operazione di divisione gli operandi *possono* essere interi: in tal caso però il risultato *non è* intero, ma reale.
Quindi se fate

12 DIV 5 otterrete 2 (intero)

e se fate

12 / 5 otterrete 2.4 (reale)

Il tipo booleano

È noto - spero - che le variabili booleane possono assumere solo due valori: falso e vero. Ciò vale evidentemente anche per PASCAL. Così come gli operatori con cui si agisce su delle variabili logiche sono sempre i soliti AND, OR, NOT.
Ci sono però alcune novità: più precisamente *esiste una relazione* fra i valori di verità. Così, come dite che tre è minore di sette, direte anche che falso *è minore di* vero. Detto così sembra una cosa da poco, invece è un'innovazione essenziale che estende notevolmente la possibilità di lavorare con variabili booleane.
Vediamo perché, e prendiamo l'operazione di implicazione B => A (B ⊃ A). Questa vuol dire che "se B è vero, allora è vero anche A", (però evidentemente B *può* essere falso ed A restare vero). Chiariamo la faccenda con un diagramma di Venu e con una tabella della verità (v. Figura 1).

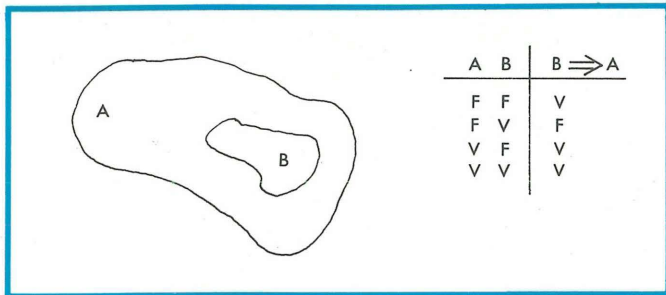



Figura 1 - Diagramma di Venu con tabella della verità.

Vi ricordo che la tabella della verità si legge in questo modo:

"Se si vuole che sia vero che la verità di B *implica* la verità di A allora occorre che:
a) se B è falso A può essere falso o vero
b) se B è vero A deve essere vero
mentre non è possibile che B sia vero ed A sia falso".

Se esaminate la tabella vedrete che (accettata la relazione F < V) abbiamo che l'operazione di implicazione si può scrivere come:

$$B \leqslant A \qquad (2.11)$$

che diventa, in PASCAL:

(2.12)

La notazione usuale (0 al posto di F; 1 al posto di V) permette di scrivere le relazioni tipo la (2.11) direttamente, dal momento che tutti sanno che 0 < 1.
Questo mi porta a parlarvi degli operatori relazionali in PASCAL. Eccoli:

= .... è identico a .... (2.13)
<> .... è diverso da ....
<= .... è minore o identico a ....
< .... è minore di .....
> .... è maggiore di .....
>= .... è maggiore o uguale a ....

Ebbene, così come abbiamo visto che l'operazione di implicazione fra due variabili logiche si può scrivere semplicemente in PASCAL (e non in altri linguaggi) proprio perché è stata introdotta la relazione F<V, potremmo far vedere che si possono scrivere ora *tutte* le operazioni possibili nell'algebra booleana senza alcuna restrizione (v. Figura 2).

Nome	Espressione usuale	Espressione in PASCAL
ZERO	0	----------
AND	$X \cdot Y$	X AND Y
INHIBIT	$X \cdot \overline{Y}$	X AND NOT Y
IDENTTITYX	X	----------
INHIBIT	$X \cdot Y$	NOT X AND Y
IDENTITY Y	Y	----------
ESCLUSIVE OR	$X + Y$	X <> Y
OR	$X + Y$	X OR Y
NOR	$\overline{X} + \overline{Y}$	NOT (X OR Y)
EQUALITY	$X \cdot Y$	X = Y
NOT Y	$\overline{Y}$	NOT Y
IMPLICATION	$X + \overline{Y}$	X <= Y
NOT X	$\overline{X}$	NOT X
IMPLICATION	$\overline{X} + Y$	X >= Y
NAND	$\overline{X \cdot Y}$	NOT (X AND Y)
ONE	1	----------

Figura 2 - Operazioni dell'algebra booleana e relativa scrittura in PASCAL.

Vedete quindi che ora tutte le possibili funzioni booleane sono implementate. Unica eccezione: le funzioni banali tipo ZERO, ONE etc.

Il tipo CHARACTER

Purtroppo qui non posso darvi altro che delle linee generali, dal momento che i caratteri non costituiscono un sistema standardizzato. In ogni caso ci sono delle richieste minime che vanno soddisfatte in ogni implementazione. Più precisamente i caratteri devono consistere almeno di:

- le lettere dell'alfabeto
- i digits da 0 a 9
- il blank

Una variabile di tipo CHAR può quindi assumere questi "valori" e su questi valori si può agire con una serie di operazioni e di funzioni. Per fissare le idee supponiamo che i caratteri constino solo delle lettere dell'alfabeto. Anzitutto bisogna notare che i caratteri vengono automaticamente *ordinati* da PASCAL: supponiamo quindi che le lettere vengano ordinate proprio in ordine alfabetico (ciò per altro non è tassativo). Ecco le funzioni che si possono far agire sui caratteri:

ORD (C): questa dà il numero ordinale corrispondente alla posizione del carattere C nell'insieme dei caratteri.

Nel nostro esempio (lettere dell'alfabeto)

ORD (A) dà 1
ORD (B) dà 2
ORD (C) dà 3 etc.

CHR (I): questa è la funzione inversa della precedente: dà come risultato il carattere che corrisponde alla posizione I.
Sempre nel nostro esempio:

CHR (1) dà A
CHR (2) dà B
CHR (3) dà C etc.

PRED (C): dà come risultato il carattere *antecedente*

(PREDecessor) al carattere C.

Eccovi alcuni esempi:

PRED (B)	dà	A	
PRED (C)	dà	B	
PRED (H)	dà	G	etc.

SUCC (C): dà come risultato il carattere *successivo* al carattere C.

Anche qui alcuni esempi:

SUCC (B)	dà	C	
SUCC (C)	dà	D	
SUCC (M)	dà	N	etc.

Queste funzioni sono implementate per il tipo CHAR ed anche per tutti i tipi che l'utente voglia definire. Oltre a queste funzioni sono validi tra i caratteri *tutti* gli operatori relazionali. Più precisamente: noi possiamo associare alle lettere dall'A alla Z i numeri da 1 a 26 (numeri ordinali). Ebbene: le stesse relazioni che valgono fra i numeri ordinali suddetti valgono anche fra le lettere corrispondenti. Per es. A è al primo posto (numero 1) e C al terzo (numero 3), e così come è vero che 1 < 1 *è anche vero che A<C.* Questa innovazione è particolarmente apprezzata quando si hanno problemi di ordinamento in banche di dati che non necessariamente sono numerici.

Errata corrige

Nella Parte I (BIT N. 5, pag. 55 e seguenti) compaiono degli errori di stampa:
L'espressione (13), a pag. 61, va corretta come segue:
<costante alfanumerica>::=<lettera>
{<digit>|
< lettera >}

A pag. 62, le metaespressioni di < soggetto > e < predicato verbale > vanno così corrette:
<soggetto>::=sostantivo>}< aggettivo>}
< sostantivo >}
<predicato verbale>::=< voce verbale >

Soluzioni degli esercizi presentati nell'articolo

CONTROLLO 1

I) manca il nome del file.
II) manca la parola PROGRAM.
III) manca il punto e virgola (;).
IV) mancano sia il punto e virgola che il nome del file.
V) giusto.
VI) giusto.
VII) mancano sia il punto e virgola che il nome del file.
VIII) "PROGRAM", non "MAIN" è la reserved word (metacostante).
IX) il nome del file va rinchiuso in parentesi.
X) dopo il nome del programma non va messa la virgola.

CONTROLLO 2

I), II), IV), VI), IX) giusti.
III), VII) non è permesso usare metacostanti come identificatori.
V), X) non è permesso l'uso di simboli diversi da digits numerici o caratteri alfabetici.
VIII) non è permesso iniziare un identificatore con dei numeri.
Per quanto concerne la domanda a parte, la risposta è sì: è possibile indicare due variabili diverse con gli identificatori VI e IX. Ciò è dovuto al fatto che PASCAL "capisce" i primi 8 digits di un identificatore e quindi (nel caso in esame) ignora semplicemente le ultime quattro lettere dell'identificatore IX (-CCIO). I due identificatori sono letti da PASCAL rispettivamente CESARINO e CESARINA e quindi possono benissimo servire ad individuare due variabili diverse.

CONTROLLO 3

I) le costanti
II) i labels.
III) le costanti.
IV) i tipi.
V) sì; anzi è obbligatorio.

CONTROLLO 4

I), II), III) giusti
IV) troppo lungo: non ci possono essere più di 4 digits.
V), VI) non sono ammesse le lettere.
IX), XII) giusti.
VII), VIII) manca il punto e virgola.
X) un label è troppo lungo.
XI) non sono ammesse lettere nei labels.

CONTROLLO 5

I), II), III), IV), V), VII), VIII), X), XI), XII), XIII), XV), XVI) giusti.
VI) come numero non funziona, e come stringa non è racchiusa in apici.
IX) è un errore comune. PASCAL pretende che si scriva 0.27 e non .27. Anche questa è un'ottima abitudine da prendere.
XIV) va scritto -22.0 E-23.

CONTROLLO 6

I) giusto (e pubblicitario).
II) manca il punto e virgola.
III)
IV) giusti.

CONTROLLO 7

I), II) giusti.
III) 2A78 è un identificatore non legale, perché inizia con un numero.
IV) è un gran pasticcio: manca un punto e virgola; BA è definito sia come carattere che come intero; CZ è definito come intero e come booleano.

Spero ve la siate cavata bene

Confronto tridimensionale BASIC-PASCAL

di **F. Del Vecchio, M. Valsasina**

I linguaggi e le possibilità grafiche dell'Apple II per realizzare un esempio pratico di comparazione fra Pascal e BASIC.

Introduzione

PASCAL, PASCAL, PASCAL; tutti ne parlano quale linguaggio del futuro: vediamolo anche sul banco di prova.
La scelta riguardante le finalità pratiche dell'esempio è ben ardua per chi vuole effettuare un confronto obbiettivo tra Pascal e BASIC.
Un programma molto grosso ci offre la possibilità di far cadere facilmente la bilancia a favore del Pascal, ma rischiamo poi superficialità nel confronto per ragioni di spazio, oppure tediose comparazioni che annoierebbero sicuramente il lettore.
Di qui la necessità di condensare in poche pagine di listings un esempio che abbia comunque utilità ed interesse.
Per soddisfare tali presupposti, ci siamo orientati verso un'applicazione grafica, in modo da dare così un supporto tangibile anche al discorso Computer Grafica del N. 5 di Bit.
A questo punto la scelta del computer sul quale sviluppare il programma non poteva che cadere sul personal Apple II, il quale, oltre al BASIC in virgola mobile, supporta anche il Pascal UCSD, ed è unanimemente riconosciuto come il personal con le migliori caratteristiche grafiche.
Non spaventi la sigla UCSD: essa indica semplicemente la versione Pascal implementata dalla Università di California, San Diego, che costituisce un sistema completo software per microcomputers standalone.
Tale versione differisce dallo Standard per un limitato numero di omissioni, per pochissime alterazioni e numerose estensioni, quali ad esempio le possibilità grafiche, ampiamente utilizzate nell'esempio che ci accingiamo a sviluppare.
È da sottolineare che l'approccio al problema è ovviamente di tipo top-down strutturato, metodologia che permette di sviluppare il programma in modo algoritmico, così come il Pascal richiede ed invoglia ad usare.
Questo è il primo beneficio che si acquisisce, cioè una mentalità strutturata anche per lo sviluppo di programmi personali su computers alla portata di tutti.

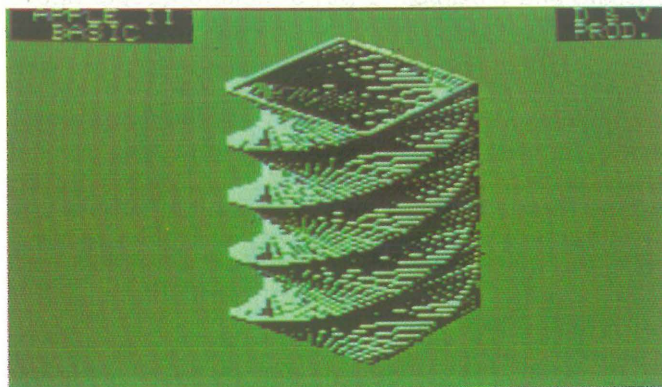
Intendiamoci sui termini

Prima di entrare in merito all'argomento specifico è bene essere a conoscenza del corretto significato di termini chiave utili a comprendere nozioni classiche di informatica ed alcune modalità operative del sistema BASIC e Pascal su Apple II.
L'APPLESOFT è il linguaggio BASIC esteso disponibile sul personal computer Apple II, che, oltre alle funzionalità tipiche del BASIC in virgola mobile, comprende anche comandi di controllo grafico (HCOLOR, HPLOT, ecc.) per disegnare rette in coordinate cartesiane.
L'APPLESOFT è un sistema completamente interattivo, è cioè possibile inserire linee, listare, eseguire il programma in qualsiasi momento.
Benché in BASIC ogni linea di sorgente sia singolarmente precompilata, cioè tradotta in un codice intermedio, l'esecuzione è di tipo interpretativo. Viene cioè effettuato un riconoscimento e controllo globale durante la fase di esecuzione del programma.
In BASIC l'unità elementare di programmazione è il programma, non è cioè possibile una suddivisione in unità modulari separate; l'unica possibilità è costituita dal CHAIN, cioè esecuzione sequenziale di più programmi concatenati fra loro.
Il Pascal è un linguaggio che richiede la compilazione globale del programma, cioè la traduzione completa delle informazioni fornite alla macchina da formato sorgente in codice eseguibile.[1]. L'entità compilabile viene identificata come modulo, o in termini elementari come mattone.
In Pascal è possibile eseguire il link, cioè unire più moduli per costruire il programma completo. Per l'Apple II i moduli linkati possono essere sia PASCAL sia ASSEMBLER.

Data la ridotta dimensione del programma esempio, non è stata utilizzata l'opzione di link.
Possiamo individuare quindi le seguenti fasi di realizzazione del proramma sul sistema Pascal:

- *Editing:* digitazione completa del programma o correzione, che il sistema Apple II supporta con editor di pagina molto efficiente e sofisticato;
- *Compile:* traduzione del sorgente e controllo formale. Su Apple II è possibile anche la correzione interattiva con posizionamento automatico sulla frase, nel punto in cui si verifica errore;
- *Link:* fusione dei moduli in un unico programma eseguibile;
- *Debug:* prova del programma con ricerca di eventuali errori.

Considerazioni generali

La nascita di linguaggi di programmazione sempre più evoluti e potenti rispecchia la necessità di avere a disposizione strumenti di lavoro che siano contemporaneamente semplici, potenti, logici, ed inoltre che permettano una metodologia di sviluppo del software che ottimizzi il tempo necessario alla realizzazione di un programma.
Il Pascal è la logica conseguenza dello studio di un linguaggio che rispetti questi presupposti. Infatti è studiato in modo da facilitare la programmazione strutturata, cioè l'approccio di tipo top-down al pro-

Prisma a 4 lati - incidenza gradi 45 - rotazione gradi 0 torsione gradi 360

[1] *Più esattamente la compilazione produce un codice intermedio, chiamato P-code, che viene eseguito tramite interpretatore.*

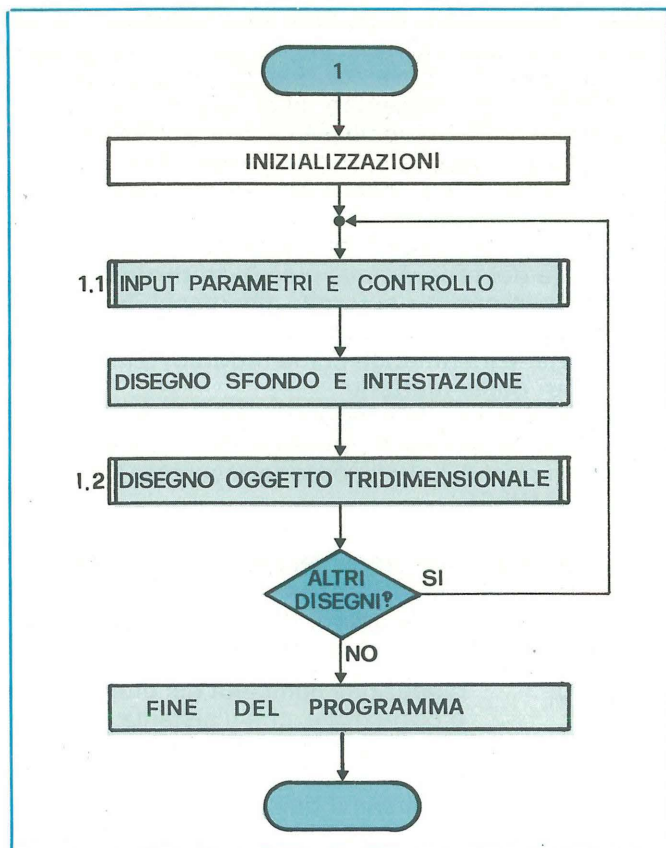



Figura 1 - Programma principale. I numeri indicano le procedure espanse successivamente.

blema, e la modularizzazione del programma (cioè la suddivisione di esso in vari sottoprogrammi che possono anche essere preparati separatamente, per poi essere uniti con una fase di link in un programma unico).

L'approccio algoritmico e strutturato ad un problema rende il programma facilmente comprensibile, cioè leggibile, anche a chi non lo ha scritto, per cui un programma in Pascal si autodocumenta notevolmente, riducendo la necessità di aggiungere al progamma documentazione accessoria.
La possibilità di modularizzare i programmi presenta anche il vantaggio di poter utilizzare le stesse procedure in diversi programmi, evitando di riscriverle ogni volta; inoltre, avendo la necessità di modificare una parte del programma, non è necessario rifare la compilazione di tutto, ma solo delle procedure interessate alla modifica, e poi di linkare queste nuove procedure nel programma principale.

Quindi il Pascal, poiché offre un adeguato repertorio di strutture di controllo e di tipi di dati, permette al programmatore di restare ad un alto livello di astrazione algoritmica, laddove invece linguaggi come il Basic lo forzano ad intaccare la logica della soluzione, con trucchi di programma e salti improvvisi che ne riducono la chiarezza. Cosicché il Pascal esprime gli algoritmi chiaramente e semplicemente, mentre il BASIC ottiene gli stessi risultati ma in un modo difficilmente comprensibile.
La notevole differenza tra la potenzialità del Pascal e quella del BASIC è rilevabile dalla Tabella I, in cui sono riportati i vari tipi di dichiarazioni e di frasi permesse dall'uno e dall'altro linguaggio.

Analisi dell'applicazione

Il programma qui riportato permette di visualizzare figure solide tridimensionali, con caratteristiche definite dall'operatore, che può scegliere:
- tipo di solido (prisma, piramide, sfera, emisfera, iperboloide);
- numero dei lati della figura base;
- angolo di incidenza;
- angolo di rotazione;
- angolo di torsione.

Questa realizzazione non è fine a se stessa, poiché la routine base del programma, che disegna un poligono regolare secondo una angolazione a scelta, può essere utilizzata come nucleo di applicazioni anche molto più complesse.

L'approccio strutturato al problema permette subito di mettere in risalto quali sono le fasi essenziali del programma e quindi quali procedure devono comporlo. Infatti nel Flow 1 (v. Figura 1) si vedono chiaramente cinque fasi:
- inizializzazione
- imput e controllo dei parametri
- disegno dello sfondo e delle intestazioni
- disegno dell'oggetto
- fine del programma.

A questo punto il problema è già stato completamente inquadrato nelle sue componenti (top), per cui si può cominciare a sviluppare le procedure che sono state individuate (down).
Il Flow 1.1 (v. Figura 2) rappresenta la routine di im-

Tabella I - Confronto tra le dichiarazioni e le frasi in Pascal e in BASIC, con riferimento all'Apple II.

	BASIC	PASCAL
- Strutture di controllo		
if	x	x
case		x
for	x	x
while		x
repeat		x
goto	x	x
- Sottoprogrammi		
recursività		x
variabili locali		x
parametri		x
- Tipi di dati		
integer	x	x
real	x	x
character e string	x	x
boolean		x
defined scalar (type)		x
- Strutture di dati		
array	x	x
record		x
set		x
file		x
pointer		x
- Operatori		
aritmetici	x	x
divisione intera	x	x
modulo	x	x
esponenziazione	x	
operatori relazionali	x	x
operatori su set		x
operatori logici	x	x

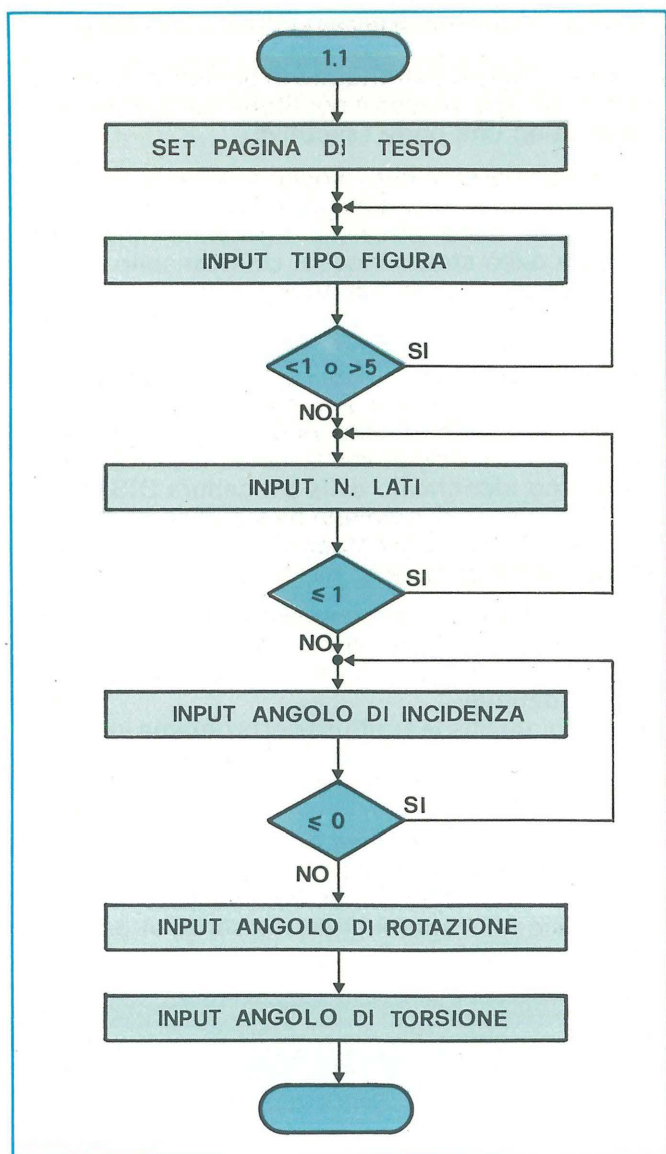



Figura 2 - Input parametri e controllo.

put dei parametri necessari al disegno ed il controllo dei valori introdotti.
Nel Flow 1.2 (v. Figura 3) si vede che la metodologia di analisi top-down viene recursivamente applicata anche scendendo di livello di astrazione.
Ci sembra doverosa a questo punto una premessa, fatta in tutta umiltà: siamo solo dilettanti nel disegno in assonometria, e la soluzione algoritmica utilizzata non è certo di tipo professionale.
I risultati ci hanno però entusiasmato a tal punto che non abbiamo voluto approfondire l'argomento. Inoltre significativa è per il lettore la semplicità con cui è stato analizzato il plottaggio del solido; suddividendolo in tante figure piane, proprio come l'approccio top-down richiede.

Supponendo di utilizzare una procedura di disegno di figura piana in assonometria, basta richiamare questa procedura più volte cambiando i parametri per ottenere gli effetti richiesti.
Cambiando la coordinata Y0 si può ad esempio costruire semplicemente il solido per accrescimento, ed inoltre variando l'angolo di rotazione si può ottenere la torsione.

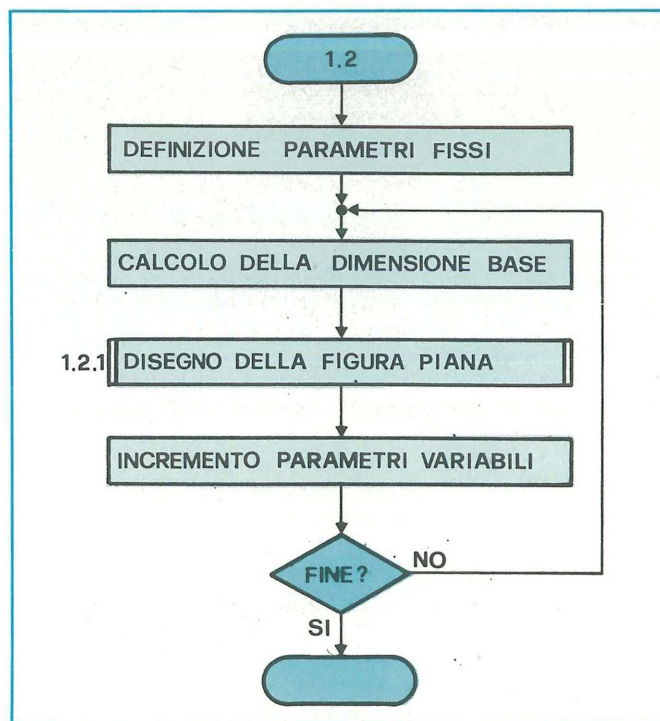



Figura 3 - Disegno dell'oggetto tridimensionale. La procedura di disegno della figura piana (1.2.1.) è espansa in Figura 4.

Il disegno di una figura piana regolare in assonometria è stato risolto inscrivendo il poligono in un'ellisse, la cui ellitticità dipende dall'angolo di incidenza (cioè di vista).
Inscrivere un poligono regolare in un'ellisse è veramente semplice: basta tracciare delle linee fra i punti

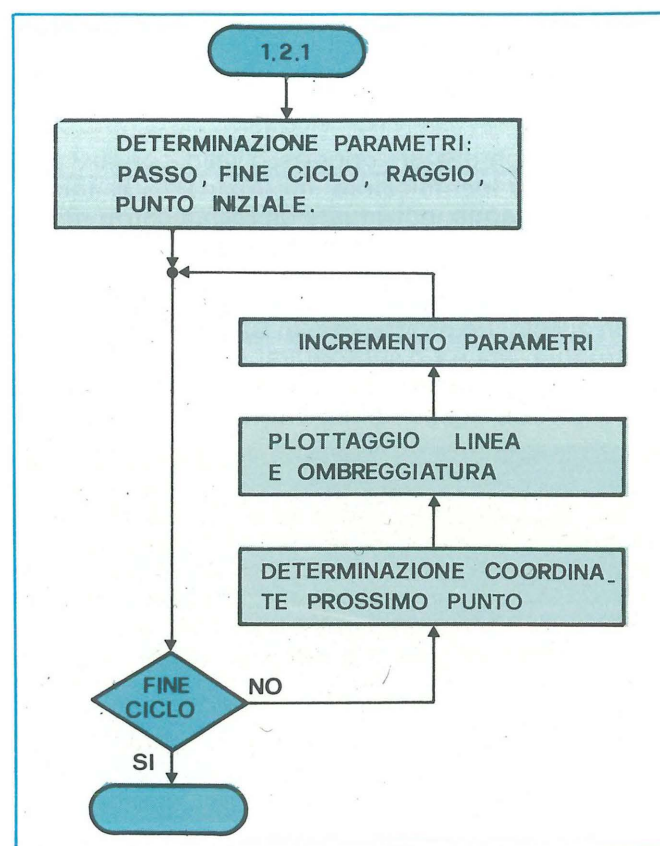
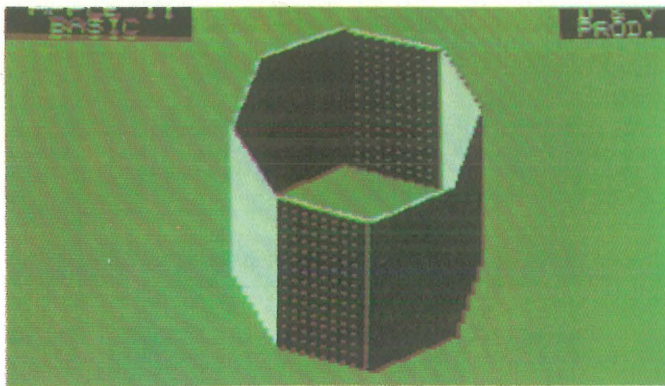
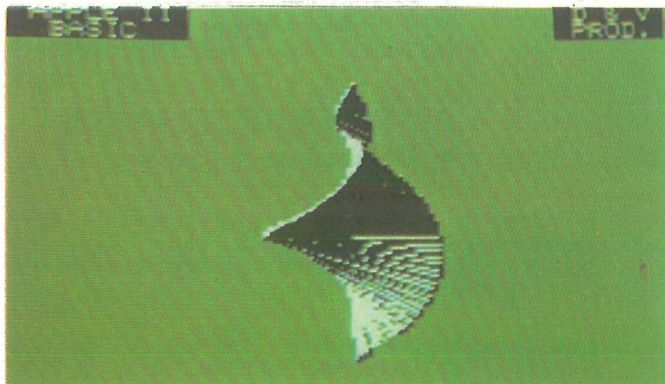



Figura 4 - Disegno della figura piana.

Prisma a 8 lati - incidenza gradi 35 - rotazione gradi 5 torsione gradi 0.

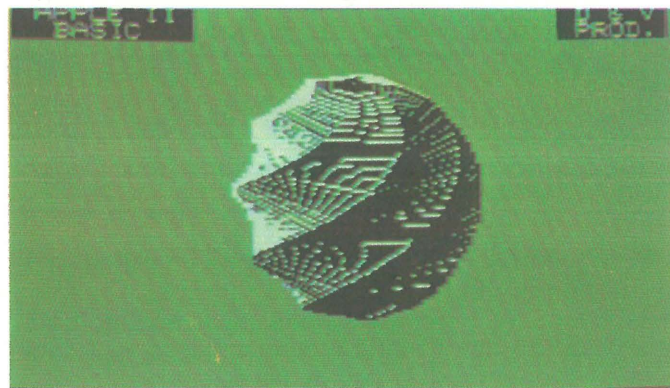
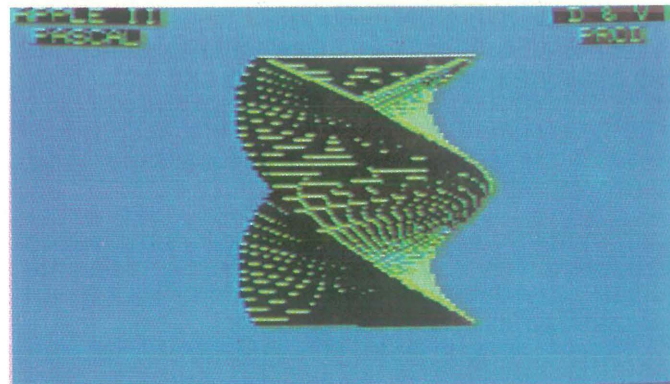
di intersezione dell'ellisse con radiali di apertura angolare 360 diviso il numero di lati (Flow 1.2.1 in Figura 4).
Semplice! Proprio per dilettanti.
Come si può notare, per facilitare il confronto tra i listings, nello sviluppo della codifica Pascal e BASIC abbiamo utilizzato gli stessi nomi per le variabili, e uguali commenti.
Per tale motivo vi sono nomi di variabili poco identificabili a causa di limitazioni BASIC che abbiamo dovuto imporre anche al Pascal.

Analisi comparativa

Vogliamo ora confrontare punto per punto le due realizzazioni.
- *Identificatori*
In BASIC gli identificatori (nomi di variabili) possono essere di lunghezza a scelta, ma solo i primi due caratteri sono riconosciuti dall'interprete, per cui ad esempio le due variabili NUMBER e NUMERATION, pur essendo diverse, vengono considerate uguali, cosicché bisogna prestare molta attenzione alla scelta dei nomi.
In Pascal il compilatore riconosce validi i primi 8 caratteri di ogni identificatore. Inoltre in Pascal anche le procedure sono individuate da identificatori mnemonici invece che da numeri come in BASIC.
- *Struttura del programma*
La maggior differenza tra i due linguaggi risiede proprio in questo: un programma in BASIC non ha alcuna struttura, ma è un susseguirsi di frasi tutte dello stesso valore, e questo comporta un aumento delle difficoltà di debug, modifica e documentazione.

Piramide a 2 lati - incidenza gradi 65 - rotazione gradi 180 torsione gradi 270.

In Pascal l'algoritmo è messo in risalto nel programma principale, che si trova in fondo al listing. Un programma in Pascal è strutturato in blocchi uno interno all'altro, ed ogni blocco è costituito da una parte dichiarativa ed una parte esecutiva.

Nella prima parte di ogni blocco vi sono le dichiarazioni delle variabili, procedure e funzioni che saranno utilizzate dalle istruzioni contenute nella parte esecutiva dello stesso blocco. Una variabile dichiarata in un blocco ha significato solo all'interno di quel blocco (è locale in esso); le variabili definite in un blocco più esterno sono però utilizzabili nel blocco interno (cioè sono variabili globali per il blocco interno), a meno che non vengano ridefinite in esso, nel qual caso il riferimento valido è quello locale.
Ad esempio nel programma la variabile BASE è locale al blocco identificato dalla procedura DISEGNO, quindi all'esterno di esso non ha significato; ma è utilizzata nel blocco della procedura PIANA che è al suo interno, cioè è globale ad esso.

Inoltre la variabile I del blocco DISEGNO viene ridefinita nel blocco PIANA, per cui all'interno di PIANA la variabile I esterna non esiste più, avendone una propria ridefinita.
Si vede quindi che la struttura del programa in Pascal procede dal generale al particolare, seguendo perfettamente la metodologia top-down di analisi del problema.
Mentre tale struttura gerarchica è dal Pascal imposta, con impegno abbiamo cercato di riprodurla anche in BASIC, arricchendo di documentazione (REM) e suddivisione in subroutines, ma senza ottenere risultati soddisfacenti.

Sfera a 4 lati - incidenza gradi 65 - rotazione gradi 56 torsione gradi 130.

Iperboloide a 6 lati - incidenza gradi 60 - rotazione gradi 0 torsione gradi 0.

- Tipi di dati e di strutture

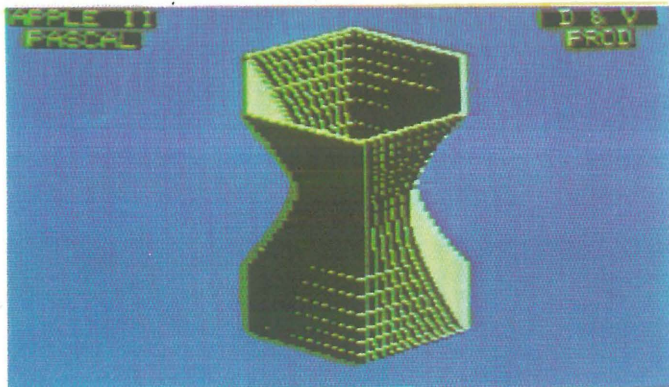
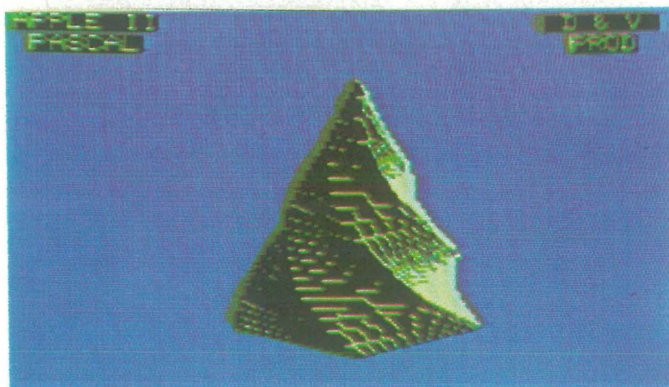
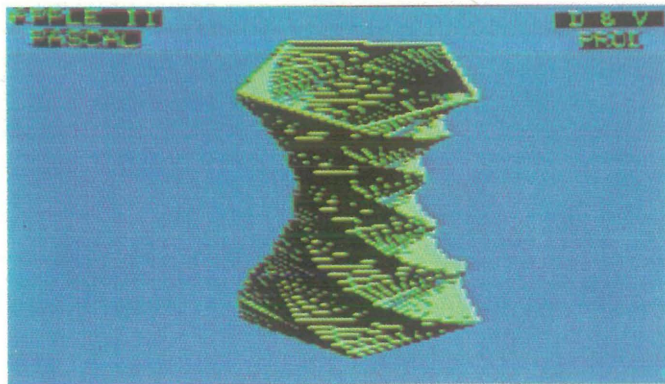
Come si vede anche dalla Tabella I, il Pascal permette l'uso di un maggior numero di tipi di dati e di strutture, semplificando notevolmente la trattazione. Ad esempio nella routine di imput parametri viene utilizzata la struttura Pascal SET per fare il controllo di range sui dati introdotti da tastiera, mentre in BASIC si devono utilizzare più test.

Nella procedura ALTRI si utilizza una variabile (FINE) di tipo logico, che può avere quindi i valori vero o falso. In Pascal è inoltre possibile definire il tipo di una variabile, cioè definire un nuovo insieme di elementi che abbiano una caratteristica propria. Ad esempio nel programma viene definito un tipo Screencolor e i valori possibili per le variabili di quel tipo (i colori del video); quindi si possono poi utilizzare quelle variabili alla stessa stregua degli altri tipi (cioè i real, integer, ecc.).

-Strutture di controllo

Confrontiamo ora i linguaggi nelle strutture di controllo, cioè l'insieme delle frasi che guidano l'esecuzione del programma.

Ad esempio nella procedura DISEGNA la dimensione della figura viene scelta in funzione della variabile TIPO: tutto ciò è ben visibile ed immediatamente comprensibile in Pascal, dove si è utilizzata la frase CASE OF, mentre in BASIC l'uso dello statement ON GOSUB è certamente meno chiaro.

Per quanto riguarda le strutture ripetitive, il BASIC ha solo lo statement FOR ... NEXT con indice del ciclo calcolato, mentre nel programma Pascal compaiono anche gli statements "WHILE DO" e "REPEAT UNTIL", in cui i test di fine ciclo possono anche essere logici.

Prisma a 3 lati - incidenza gradi 75 - rotazione gradi 60 torsione gradi 130.

Piramide a 5 lati - incidenza gradi 70 - rotazione gradi 0 torsione gradi 180.

Iperboloide a 5 lati - incidenza gradi 65 - rotazione gradi 0 torsione gradi 360.

Per realizzare un ciclo su condizione logica in BASIC, quindi, bisogna utilizzare lo statement FOR ... NEXT con all'interno un GOTO ad uno statement esterno al ciclo per condizione verificata.

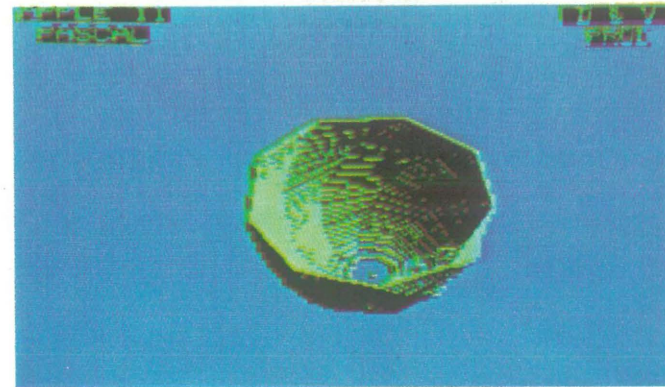
- Procedure e funzioni

Abbiamo già visto come in Pascal le procedure e le funzioni sono identificate non da un numero di linea ma da un nome.

Inoltre è possibile scrivere routines parametriche, cioè procedure in cui alcune variabili non hanno valore proprio, bensì vengono attualizzate al momento della chiamata. Un esempio è la procedura MOVETO utilizzata per il plottaggio della linea, a cui si devono passare come parametri le coordinate X e Y dal punto finale della linea da tracciare.

In BASIC tutte le variabili hanno solo significato globale. Nel listing Pascal compaiono inoltre le dichiarazioni di procedure, che però non sono sviluppate nel programma: sono procedure esterne che vengono incluse nel codice oggetto dal compilatore quando riconosce la frase USES.

L'esempio non è organizzato per mettere in risalto tutte le peculiarità dei linguaggi. Comunque senza alcun dubbio, anche forzando una ricca e completa documentazione nel listato BASIC, il Pascal risulta più leggibile e lineare, riproducendo in codifica la stessa struttura dettata dai flow di analisi.

Lasciamo al lettore una comparazione più accurata seguendo punto per punto parallelamente i due listati (v. Tabella II per il significato delle variabili utilizzate).

Sfera a 9 lati - incidenza gradi 45 - rotazione gradi 0 torsione gradi 180.

```
    (*************************************)
(****      A P P L E    I I          ****)
(****  D  &  V    PRODUCTIONS        ****)
(****                                ****)
(****      PROGRAMMA   ESEMPIO       ****)
(****        PER DISEGNO DI          ****)
(****        POLIGONI SOLIDI         ****)
(****                                ****)
(****  MILANO  1/1/1980              ****)
(*************************************)


PROGRAM ASSONOMETRIE;
  USES TURTLEGRAFICS,TRANSCEND;          (* DICHIARAZIONE DELLE ROUTINE DI
                                            UTILITA' PER GRAFICI E PER
                                            FUNZIONI TRIGONOMETRICHE *)
  TIPE SCREENCOLOR = (NONE,WHITE,BLACK,REVERSE,RADAR,BLACK1,GREEN,VIOLET,
                      WHITE1,BLACK2,ORANGE,BLUE,WHITE2);


 CONST  X0 = 140;

 VAR    CDC,NL,AI,
        RO,TR,H        :REAL;

        Y0,TIPO,AIN    :INTEGER;

        TAS            :CHAR;

        FINE           :BOOLEAN;

        STRINGA        :STRING;


 PROCEDURE SELEZIONE(SEL:INTEGER);                    (*SELEZIONA TIPO FIGURA*)

   BEGIN
   CASE SEL OF
     1: STRINGA:='PRISMA';
     2: STRINGA:='PIRAMIDE';
     3: STRINGA:='SFERA';
     4: STRINGA:='EMISFERA';
     5: STRINGA:='IPERBOLOIDE';
   END;
   END;                                 (*SELEZIONE*)


 PROCEDURE PARAMETRI;                   (* INPUT DEI PARAMETRI *)

   VAR J               :INTEGER;
       CONT            :STRING;


 BEGIN
   TEXTMODE;                            (*SET PAGINA DI TESTO*)
   UNITCLEAR(1);                        (*POSIZIONA CURSORE INIZIO PAGINA*)
   GOTOXY(0,1);                         (*INPUT TIPO DI FIGURA*)
   WRITELN('TIPO FIGURA =');
   J:=1;
   REPEAT
     SELEZIONE(J);
     GOTOXY(8,J+2);
     STR(J,CONT);
     WRITELN(CONCAT(CONT,'  ',STRINGA));
     J:=J+1;
     UNTIL J>5;

   REPEAT
     GOTOXY(12,9);  WRITE('QUALE ? ');
     READLN(TIPO);
     UNTIL TIPO IN [1..5];

   REPEAT                               (* INPUT NUMERO LATI *)
     GOTOXY(0,12);
     WRITE('NUMERO LATI = ');
     READLN(NL);
     UNTIL  NL  > 1;

   REPEAT                               (* INPUT ANGOLO DI INCIDENZA *)
     GOTOXY(0,15);
     WRITE('ANGOLO DI INCIDENZA = ');
     READLN(AIN);
     UNTIL  AIN IN [1..90];

   GOTOXY(0,18);                        (* INPUT ANGOLO DI ROTAZIONE *)
   WRITE('ANGOLO DI ROTAZIONE = ');
   READLN(RO);

   GOTOXY(0,21);                        (* INPUT ANGOLO DI TORSIONE *)
   WRITE('ANGOLO DI TORSIONE = ');
   READLN(TR);

 END;                                   (* PARAMETRI *)


 PROCEDURE SFONDO;                      (* SET CAMPO GRAFICO *)

 VAR LATI,ANGOLI   : STRING;

 BEGIN
   INITTURTLE;                          (* INIZIALIZZA GRAFICA *)
   VIEWPORT(0,279,31,191);              (* SET FINESTRA GRAFICA *)
   FILLSCREEN(ORANGE);
   VIEWPORT(0,279,0,191);               (* RIPRISTINA INTERO VIDEO *)
   CHARTYPE(10);                        (* SCRITTURA INTESTAZIONI *)
   PENCOLOR(NONE);
   MOVETO(5,181);
   WSTRING('APPLE II');
   MOVETO(219,181);
   WSTRING(' D & V ');
   MOVETO(230,171);
   WSTRING('PROD');
   MOVETO(12,171);
   WSTRING('PASCAL');

   MOVETO(0,21);                        (* SCRITTURA PARAMETRI *)
   SELEZIONE(TIPO);
   STR(AIN,ANGOLI);
   STR(ROUND(NL),LATI);
   STRINGA:=CONCAT(STRINGA,' ',LATI,' LATI  INCIDENZA GRADI ',ANGOLI);
   WSTRING(STRINGA);
   MOVETO(0,10);
   STR(ROUND(RO),LATI);
   STR(ROUND(TR),ANGOLI);
   STRINGA:=CONCAT('ROTAZIONE GRADI ',LATI,'  TORSIONE GRADI ',ANGOLI);
     WSTRING(STRINGA);

 END;                                   (*SFONDO*)


 PROCEDURE DISEGNO;                     (*DISEGNO FIGURA SOLIDA*)

 VAR   Y0, BASE,CAI,
        I,PASSO        : REAL;


   PROCEDURE PIANA;                     (*DISEGNO FIGURA PIANA*)

   VAR GIRO,I,RAGGIOY: REAL;
       PX,PY,X,Y       : INTEGER;

   BEGIN
     GIRO:=8*ATAN(1)+RO+0.001;          (*NUMERO ITERAZIONI*)
     RAGGIOY:=BASE*CAI;                 (*DIMENSIONE SU ASSE Y*)
     PX:=ROUND(X0+BASE*SIN(RO));        (*X PRIMO VERTICE*)
     PY:=ROUND(Y0+RAGGIOY*COS(RO));     (*Y PRIMO VERTICE*)
     PENCOLOR(NONE);
     MOVETO(PX,PY);                     (*POSIZIONAMENTO PRIMO VERTICE*)
     I:=RO;
     WHILE I <= GIRO DO                 (*CICLO DISEGNO*)
     BEGIN                              (*FIGURA PIANA*)
       Y:=ROUND(Y0+RAGGIOY*COS(I));     (*Y PROSSIMO VERTICE*)
       X:=ROUND(X0+BASE*(SIN(I)));      (*X PROSSIMO VERTICE*)
       PENCOLOR(NONE);
       MOVETO(PX+1,PY-1);
       PENCOLOR(BLACK2);
       MOVETO(X+1,Y-1);                 (*DISEGNO OMBREGGIATURA*)
       PENCOLOR(NONE);
       MOVETO(PX,PY);
       PENCOLOR(WHITE2);
       MOVETO(X,Y);                     (*DISEGNO LATO FIGURA*)
       PX:=X;PY:=Y;
       I:=I+PASSO;
     END;
   END;                                 (*PIANA*)


 BEGIN
   RO:=(RO+180)*CDC;                    (*ANGOLO DI ROTAZ INIZIALE*)
   PASSO:=8*ATAN(1)/NL;
   AI:=AIN*CDC;                         (*ANGOLO DI INCIDENZA*)
   H:=100*SIN(AI);                      (*ALTEZZA DELLA FIGURA*)
   Y0:=112-ROUND(H / 2);                (* Y CENTRO FIGURA *)
   CAI:=COS(AI);
   I:=0;

   REPEAT
     CASE TIPO OF
       1: BASE:=50;                     (*DIMENSIONE PRISMA*)
       2: BASE:=50*(1-I/100);           (*DIMENSIONE PIRAMIDE*)
       3: BASE:=50*SQRT(1-SQR(1-I/50));(*DIMENSIONE SFERA*)
       4: BEGIN
            BASE:=50*SQRT(1-SQR(I/50));(*DIMENSIONE EMISFERA*)
            IF I=50 THEN I:=110;
          END;
       5: BASE:=50*SQRT(0.25+3*         (*DIMENSIONE IPERBOLOIDE*)
                    SQR(I/100-0.5));
       END;

     RO:=RO+TR*CDC/100;                 (* CALCOLO ROTAZIONE *)
      PIANA;                            (* DISEGNA UNA FIGURA PIANA *)
      Y0:=Y0+H/100;
      I:=I+1;
      UNTIL I>100;
   END;                                 (* DISEGNO *)


   PROCEDURE ALTRI;                     (*RICHIESTA DI ALTRI DISEGNI*)

   BEGIN
     REPEAT
       READ(TAS);
       UNTIL TAS IN ['S','N'];
     IF TAS = 'N' THEN FINE:=TRUE
                  ELSE FINE:=FALSE;
   END;


 BEGIN                                  (* INIZIO DEL PROGRAMMA PRINCIPALE *)

   CDC:=ATAN(1)/45;                     (* COSTANTE DI CONVERSIONE DA
                                           GRADI A RADIANTI *)
   REPEAT
     PARAMETRI;                         (* INPUT DEI PARAMETRI *)
     SFONDO;                            (* SET CAMPO GRAFICO *)
     DISEGNO;                           (* SOLIDO IN PROSPETTIVA *)
     ALTRI;                             (* RICHIESTA ALTRI DISEGNI *)
     UNTIL FINE;
     TEXTMODE;                          (* FINE PROGRAMMA *)
     UNITCLEAR(1);
   GOTOXY(10,23);
   WRITELN('GRAZIE PER L''ATTENZIONE');
   WRITELN('              D  &  V    PRODUCTIONS');
 END.
```

Listato del programma di generazione di figure solide in Pascal.

```
1000 REM ************************
1005 REM ** A P P L E   I I **
1010 REM ** D & V  PRODUCTION **
1020 REM ************************
1030 REM *  PROGRAMMA ESEMPIO  *
1040 REM *  PER   DISEGNO  DI  *
1050 REM *  POLIGONI   SOLIDI  *
1060 REM ************************
1070 REM *    MILANO 1/1/1980   *
1080 REM ************************
1090 REM
2000 REM ***INIZIALIZZAZIONE****
2010 REM  LOAD HIGH-RESOLUTION
2020 REM  CHARACTER GENERATOR
2030 REM  PER DICITURE FISSE
2040 D$ =  CHR$ (4): REM CTRL-D
2050  PRINT D$;"BLOAD HI-RES CHAR
     ACTER GENERATOR"
2060  PRINT D$;"BLOAD CHARACTER T
     ABLE"
2070  REM CONVERSIONE GRADI RAD.
2080 CDC =  ATN (1) / 45
2090  REM
2100  REM **INPUT DEI PARAMETRI**
2110  GOSUB 6000
2120  REM
2200  REM ***SET CAMPO GRAFICO***
2210  GOSUB 5000
2220  REM
2300  REM **PROSPETTIVA  SOLIDO**
2310  GOSUB 4000
2320  REM
2400  REM *TEST:  ALTRO DISEGNO?*
2410  GET TA$
2420  IF TA$ = "S" THEN 2100
2430  IF TA$ <  > "N" THEN 2410
2440  REM
2500  REM *** FINE  PROGRAMMA ***
2510  PR# 0: TEXT : HOME
2520  PRINT "GRAZIE PER L'ATTENZI
     ONE."
2530  PRINT  TAB( 20);"D & V  PRO
     DUCTIONS"
2540  END
3000  REM ************************
3010  REM ****  SUBROUTINE  *****
3020  REM ** PLOT FIGURA PIANA **
3030  REM ************************
3060  REM  NUM. ITERAZIONI
3070 GIRO = 8 *  ATN (1) + RO + 0
     .001
3080  REM  DIM. SU ASSE Y
3090 RAGGIOY = BASE * CAI
3100  REM  X PRIMO VERTICE
3110 PX = X0 + BASE *  SIN (RO)
3120  REM  Y PRIMO VERTICE
3130 PY = Y0 + RAGGIOY *  COS (RO
     )
3200  REM  CICLO DISEGNO
3210  REM  FIGURA PIANA
3220  FOR I = RO TO GIRO STEP PAS
     SO
3230 Y = Y0 + RAGGIOY *  COS (I)
3240 X = X0 + BASE *  SIN (I)
3250  REM  PLOT OMBRA
3260  HCOLOR= 0
3270  HPLOT PX + 1,PY + 1 TO X +
     1,Y + 1
3280  REM  PLOT LATO
3290  HCOLOR= 3
3300  HPLOT PX,PY TO X,Y
3310 PX = X:PY = Y
3320  NEXT I
3330  RETURN
4000  REM ************************
4010  REM ****  SUBROUTINE  *****
4020  REM **  DISEGNO  SOLIDO  **
4030  REM ************************
4040  REM  ANGOLO DI ROTAZIONE
4050 RO = CDC * RO
4080  REM  ALTEZZA CALCOLATA
4090 H = 100 *  SIN (AI * CDC)
4100  REM  POSIZIONAMENTO
4110 Y0 = 80 + H / 2
4120 X0 = 140
4200  REM  CALCOLO PARAMETRI
4210  REM  COSTANTI PER
4220  REM VELOCIZZARE IL DISEGNO
4230 CAI =  COS (AI * CDC)
4240 PASSO = 8 *  ATN (1) / NL
4300  REM CICLO DISEGNO SOLIDO
4310  FOR J = 0 TO 100
4320  ON TIPO GOSUB 4910,4920,493
     0,4940,4950
4324  REM CALCOLO ROTAZIONE
4326 RO = RO + TR * CDC / 100
4330  REM DISEGNO FIGURA PIANA
4340  GOSUB 3000
4350  REM CENTRO FIGURA SEGUENTE
4370 Y0 = Y0 - H / 100
4380  NEXT J
4390  RETURN
4910  REM  DIMENSIONE PRISMA
4915 BASE = 50: RETURN
4920  REM  DIMENSIONE PIRAMIDE
4925 BASE = 50 * (1 - J / 100): RETURN
4930  REM  DIMENSIONE SFERA
4935 BASE = 50 *  SQR (1 - (1 - J
     / 50) ^ 2): RETURN
4940  REM  DIMENSIONE EMISFERA
4945 BASE = 50 *  SQR (1 - (J / 5
     0) ^ 2)
4947  IF J = 50 THEN J = 110: RETURN
4948  RETURN
4950  REM DIMENSIONE IPERBOLOIDE
4955 BASE = 50 *  SQR (0.25 + 3 *
     ((J / 100) - 0.5) ^ 2): RETURN
5000  REM ************************
5010  REM ****  SUBROUTINE  *****
5020  REM ***SET CAMPO GRAFICO***
5030  REM ************************
5040  REM  INIZIALIZZA GRAFICA
5050  HGR2
5060  REM  PREPARA SFONDO
5070  HCOLOR= 1
5080  FOR I = 0 TO 159
5090  HPLOT 0,I TO 279,I
5100  NEXT I
5200  REM  SCRITTURA INTESTAZIONI
5210  REM  SET TESTO IN PAGINA
5220  REM GRAFICA HIGH-RESOLUTION
5230  POKE 54,0: POKE 55,96
5240  POKE  - 16300,0
5300  REM  DICITURE FISSE
5310  VTAB (1): HTAB (2)
5320  PRINT " APPLE II "
5330  VTAB (1): HTAB (33)
5340  PRINT " D & V "
5350  VTAB (2): HTAB (2)
5360  PRINT "  BASIC   "
5370  VTAB (2): HTAB (33)
5380  PRINT " PROD. "
5400  REM  SCRITTURA PARAMETRI
5410  RESTORE
5420  FOR I = 0 TO TIPO - 1
5430  READ TIPO$
5440  NEXT I
5450  VTAB (22): HTAB (1)
5460  PRINT TIPO$;
5470  HTAB (13): PRINT NL;
5480  PRINT "  LATI";
5490  PRINT "  INCIDENZA GRADI ";
     AI
5500  VTAB (24): HTAB (1)
5510  PRINT "ROTAZIONE GRADI ";RO
5520  PRINT "   TORSIONE GRADI ";
     TR;
5999  RETURN
6000  REM ************************
6010  REM ****  SUBROUTINE  *****
6020  REM *** INPUT PARAMETRI ***
6030  REM ************************
6040  REM SETTA PAGINA DI TESTO
6050  PR# 0: TEXT
6060  REM CURSORE INIZIO PAGINA
6070  HOME
6100  REM  INPUT TIPO FIGURA
6110  VTAB (1): HTAB (1)
6120  PRINT "TIPO FIGURA ="
6130  VTAB (3): HTAB (8)
6140  PRINT "1  PRISMA"
6150  HTAB (8)
6160  PRINT "2  PIRAMIDE"
6170  HTAB (8)
6180  PRINT "3  SFERA"
6190  HTAB (8)
6200  PRINT "4  EMISFERA"
6210  HTAB (8)
6220  PRINT "5  IPERBOLOIDE"
6230  VTAB (9): HTAB (12)
6240  INPUT "QUALE ? ";TIPO
6250  IF (TIPO < 1) OR (TIPO > 5)
     THEN 6230
6300  REM  INPUT NUMERO LATI
6310  VTAB (12): HTAB (1)
6320  INPUT "NUMERO LATI = ";NL
6330  IF NL < 1 THEN 6300
6400  REM  INPUT INCIDENZA
6410  VTAB (15): HTAB (1)
6420  INPUT "ANGOLO DI INCIDENZA
     = ";AI
6430  REM  CONTROLLO INCIDENZA
6440  IF (AI < 1) OR (AI > 90) THEN
     6400
6500  REM INPUT ANG. ROTAZIONE
6510  VTAB (18): HTAB (1)
6520  INPUT "ANGOLO DI ROTAZIONE
     = ";RO
6600  REM INPUT ANG. TORSIONE
6610  VTAB (21): HTAB (1)
6620  INPUT "ANGOLO DI TORSIONE =
     ";TR
6630  RETURN
7000  DATA   "PRISMA","PIRAMIDE",
     "SFERA","EMISFERA","IPERBOLO
     IDE"
```

Listato del programma di generazione di figure solide in BASIC.

Conclusione

Non abbiamo certo esaurito un confronto che pensiamo possa essere ripreso più e più volte in relazione alle varie applicazioni.

Contiamo comunque di aver suscitato interesse e chiarito, per quanto possibile, il nocciolo del confronto con un supporto pratico.

Chi del software ha fatto strumento di lavoro può dare il giusto valore ad argomentazioni di: *modularità, autodocumentazione, revisione, adattabilità, flessibilità, potenza,* che solo un linguaggio ad alto livello può offrire.

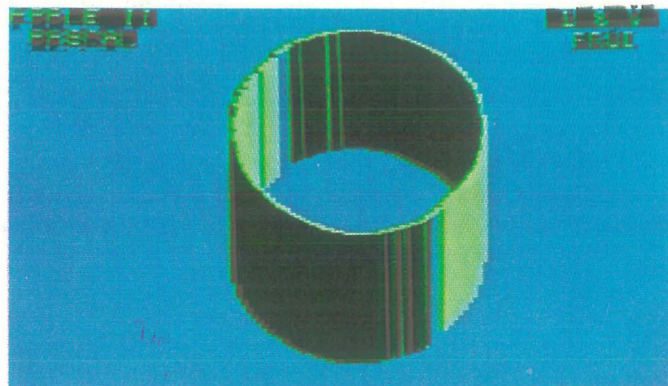
Non vogliamo però denigrare il BASIC, che rappresenta un valido ed efficiente supporto per chi voglia muovere i primi passi nella programmazione; per chi desideri approfondire la conoscenza teorica del mondo dell'informatica utilizzando il linguaggio del computer come strumento immediato per ottenere risultati altrettanto immediati.

Prisma a 32 lati - incidenza gradi 30 - rotazione gradi 30 torsione gradi 0.

Tabella II - Glossario delle variabili utilizzate nei listati.	
TIPO	TIPO di solido in cui inscivere l'assonometria desiderata. (Prisma, Piramide, Sfera, Emisfera, Iperboloide)
NL	Numero di lati della figura piana di base
AI	Angolo di incidenza, cioè di vista rispetto alla perpendicolare di giacenza del solido
RO	Angolo di rotazione del primo vertice rispetto al piano di vista verticale
TR	Angolo di torsione totale del solido
CAI	Coseno dell'angolo di incidenza
H	Altezza del solido in funzione dell'angolo di incidenza
BASE	Diametro del cerchio in cui viene inscritto il poligono regolare
X0	Ascissa del centro cerchio o ellisse
Y0	Ordinata del centro cerchio o elisse
RAGGIOY	Diagonale minore dell'ellisse in funzione dell'angolo di incidenza
PASSO	Angolo compreso fra due vertici del poligono
GIRO	Angolo di delimitazione del ciclo di disegno figura piana
PX	Ascissa dell'ultimo vertice disegnato
PY	Ordinata dell'ultimo vertice disegnato
X	Ascissa del vertice successivo
Y	Ordinata del vertice successivo
COLORE	Colore delle prossime linee di disegno
CDC	Costante di conversione gradi in radianti

Il Pascal va considerato un gradino più in alto nell'ottica del professionista informatico.
Non è da sottovalutare comunque il fatto che oggi tale linguaggio è disponibile anche su personal computers di basso costo, quindi alla portata di tutti; un'occasione in più per conoscere ed utilizzare il PASCAL.

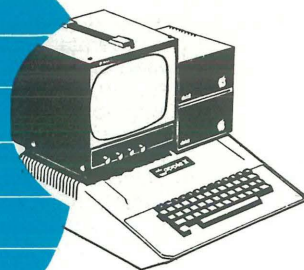
Bibliografia

1) K.Jensen, N. Wirth, *PASCAL user manual and report* (Springer-Verlag).
2) K.L. Bowles, *Problem solving using Pascal* (Springer - Verlag).
3) *Apple Pascal reference manual* (Apple Computer Inc.)
4) *Applesoft BASIC programming reference manual* (Apple Computer Inc.).
5) N. Wirth, *Principi di programmazione strutturata* (ISEDI).

LA ELIT-MICROMEGAS DI PISA FORNISCE IL GIANO, PUNTO DI PASSAGGIO PER UN SISTEMA DI INFORMATICA DISTRIBUITA.

La produzione di elaboratori in Italia vede protagoniste piccole e coraggiose società come la Elit-Micromegas, pisana. Dopo aver prodotto e venduto con successo il microcomputer ULISSE (l'unico dotato di un linguaggio di programmazione completamente sviluppato in Italia, il GAMMA), dal settembre 79 ha messo in commercio un sistema dual processor denominato GIANO (perché bifronte: le sue due CPU sono due microprocessori 8085). Le due CPU operano in parallelo, con 16 o 32 Kbyte di memoria ciascuna. Un sistema di scambio messaggi tra esse consente all'utente di vedere "trasparentemente" il GIANO come un tutto unico. Ogni CPU, indipendentemente può gestire fino a 4 diverse applicazioni provenienti da altrettanti terminali e/o sistemi Ulisse. Si tratta quindi del centro di un sistema di informatica distribuita, il cui obiettivo futuro ha già un nome, PROMETEO, che rientra in un progetto di ricerca finalizzata del C.N.R.

(G.G.)

FINALMENTE
IN ITALIA
IL PERSONAL
COMPUTER
PIU' VENDUTO
NEL MONDO

apple II

SOFTEC, Agente esclusiva di vendita dell'Apple per la Lombardia, Piemonte e Liguria, vi offre per una completa utilizzazione dell'Apple II:

- seminari di introduzione ai Personal Computer
- corsi di Basic e Pascal
- seminari su sistemi operativi a dischi (DOS, UCSD)
- assistenza e consulenza sistemistica (pre e post-vendita)
- sviluppo di software applicativo per il mercato italiano (come il potente data base ARCHIVIO)
- disponibilità dei programmi generalizzati più diffusi nel mondo (VISICALC, EASYWRITER, DESKTOP PLAN, ecc.).

Il personal computer **PROFESSIONALE** dal costo più basso (oltre 100.000 Apple II venduti nel mondo sono un risultato eloquente!)

APPLE II E' PROFESSIONALE !

- Ha le caratteristiche costruttive e di qualità dei grandi calcolatori
- è dotato di software base completo e sofisticato (monitors, sistemi operativi, linguaggi, ecc.)
- è interfacciato a tutti i tipi di periferiche (stampanti, plotters, digitizer, comunicazioni, ecc.)
- è corredato di una completa documentazione tecnica e didattica
- è l'unico sistema che vi offre il "PLUS" della grafica (vera!) e del colore.

VOLETE SAPERNE DI PIU' ?

Telefonateci o veniteci a trovare.

Agenti Apple II per:
Lombardia, Piemonte e Liguria

MILANO - via G. Govone, 56 - tel. (02)3490231
TORINO - c.so M. d'Azeglio, 60 - tel. (011)6509303/4

CERCANSI aziende qualificate per la vendita di Personal Computers nelle zone ancora libere del Piemonte, Lombardia e Liguria.

Ebbene si! Abbiamo acquistato un personal computer

di **Aldo Cavalcoli**

Convinti, come penso tutti, che mettere le mani sulla macchina vale ben più di qualsiasi disquisizione e insegnamento teorico, in un momento in cui da tutte le parti vengono proposte meraviglie con l'uso dei personal computers, è stata presa la decisione: verificare di persona su una applicazione non convenzionale.

Dopo un'attenta ed asaustiva ricerca di quanto il mercato offre (sarebbe meglio dire di quanto è realmente subito disponibile), abbiamo fatto la nostra scelta, che resta ovviamente segreta (ci scusi il costruttore).

È stato scelto un personal computers medio, volutamente medio in quanto è nostra intenzione verificare, con una struttura hardware ragionevole ma non alta, se davvero con un personal computer "si può fare quanto dicono si possa fare" (e che non è stato ancora fatto).

Posto che al lettore necessariamente non interessi il travaglio interiore del tecnico curioso e le sue scelte di investimento, resta comunque un fatto: esiste una gamma molto diversificata di settori applicativi non ancora toccati, in cui ci si può gettare. Quindi lo scopo è quello di effettuare una verifica delle possibilità applicative non convenzionali dei Personal computers, dove per convenzionale si intende quella applicazione già soddisfatta da sistemi di elaborazione dati non personal computers e ricopiata dai personal computers.

Penso ci si trovi in una classica fase di transizione molto simile a quella vissuta dalla bioingegneria ai suoi inizi, quando il gap tra esperti di medicina ed esperti di elettronica era tale che non permetteva un edeguato colloquio risolutore di situazioni efficientemente affrontabili.

Occorre "entrare nel campo", quindi far proprie le esigenze dell'utenza, con un investimento di studio che forse gli attuali computer shop non sono molto motivati a fare; sarebbe meglio dire che si tende a demanare all'utente la problematica: tu, utente, hai i problemi, questo è lo strumento, con questo strumento è stato fatto questo e questo, che un po' si avvicina alle tue esigenze, adesso datti da fare.

Nel caso di alcune categorie professionali, caratterizzate da un alto coefficiente di $, lo sforzo si sta facendo (dentisti, medici, avvocati), ma chi pensa ai singoli?

Il solito problema è mettere assieme un nucleo minimo di utenti simili e personalizzare il personal computer alle esigenze simili di tutti per poi suddividere il costo del software sulle singole unità.

Non crediamo sinceramente al cosiddetto Instant Software, made in USA, da iniettare pari pari nel personal computer acquistato, perché mai potrà andare bene al 100$, ci si potrà giusto accontantare (a volte con molta soddisfazione, peraltro).

Ed allora si è deciso di farsi carico delle problematiche dell'utente, appropriarsi delle sue procedure personali e verificare se gli conviene un personal computer.

Sarà molto interessante pubblicizzare i risultati di tale azione.

Quanti stanno per iniziare un'analoga operazione?

Lanciamo dalle pagine di BIT questo appello, nella speranza di raccogliere idee, risultati, e renderli poi pubblici per uno sviluppo culturale delle applicazioni di personal computers.

Quindi tutti coloro che fossero interessati a condividere i loro sforzi e far conoscere i loro risultati sono senz'altro invitati a scrivere alla redazione di BIT per contribuire a costruire una casistica sperimentata sulle "promesse applicative dei personal computers".

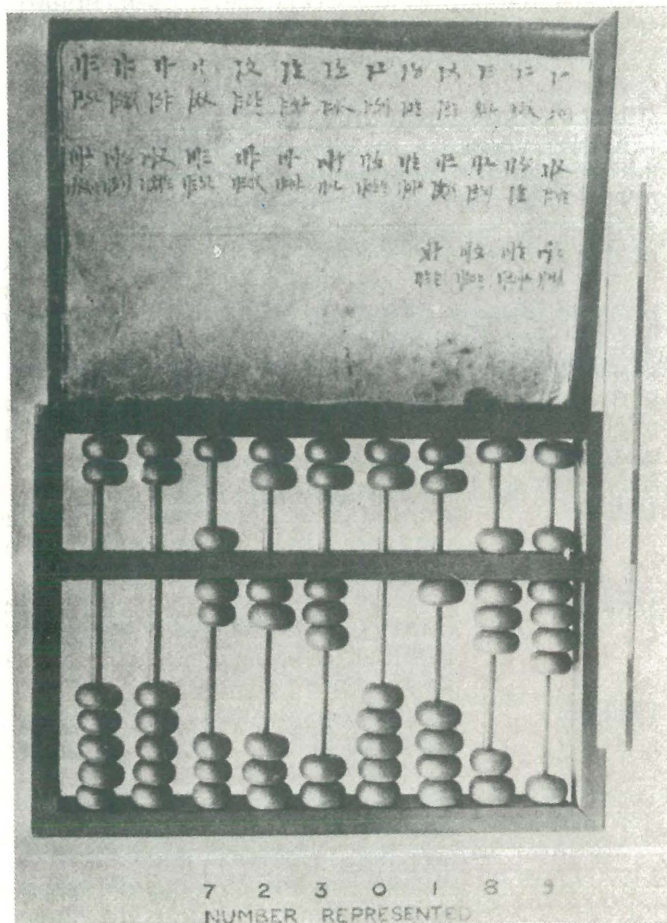
Il calcolatore per i giovani

Parte I di **F. Waldner** - *Università di Bari, Istituto di Fisica*

Premessa

Intendiamoci bene: non è affatto vietato ai meno giovani di leggere queste righe; ad un patto però: che siano dei principianti, o che - se non lo sono - non arriccino comunque il naso quando parlerò di cose ovvie e scontate. Certe cose sono infatti ovvie per qualcuno, ma per molti non lo sono affatto.
L'esigenza di articoli di questo tipo non è nata però casualmente: è invece emersa da esperienze scolastiche e redazionali, dalle quali si è visto che, vuoi a causa della rapidità del progresso tecnologico, vuoi a causa della nostra situazione scolastica, quando si parla di calcolatori e di strumentazione moderna la carenza di informazione di base si fa sentire subito. Quindi ecco degli articoli per iniziandi. Come mai questo titolo, allora? Perché penso che saranno (e sono) proprio i giovani a soffrire di più per carenze conoscitive di questo tipo, dal momento che il calcolatore entrerà sempre di più nella nostra vita e nel nostro mondo: fra vent'anni chi non saprà usare il calcolatore sarà confrontabile a colui che oggi non sa guidare l'automobile, o a colui che all'inizio del secolo non sapeva né leggere né scrivere. Troveremo i calcolatori dappertutto: negli uffici, nelle industrie, nei laboratori, in casa, attaccati al telefono, negli elettrodomestici. Conoscere l'uso del calcolatore sarà quindi il primo passo per potersi muovere nel mondo futuro in condizioni di ragionevole indipendenza e sicurezza. Questa della diffusione capillare dell'informazione e della possibilità di elaborare dei dati sarà una delle più grandi rivoluzioni tecnologiche mai avvenute: in realtà la stiamo già in parte vivendo, però il più deve ancora venire.
Questi articoli si baseranno su un doppio schema: da un lato punteranno sulle nozioni di base necessarie per comprendere il funzionamento e l'uso di un calcolatore, dall'altro sull'uso pratico di un linguaggio. A questo proposito ho pensato di scegliere il BASIC, che è estremamente diffuso soprattutto nel campo dei calcolatori da tavolo, e che probabilmente resterà in uso ancora a lungo per una vasta fascia di utenti.

Un po' di storia

Il calcolo è legato al numero, e questo alla civiltà. Tutto cominciò molto tempo addietro, quando l'uomo sentì la necessità di contare ed inventò qualcosa di più raffinato del primitivo sistema di numerazione *"uno, due, molti"*. Incidentalmente: in molte zone del mondo il sistema di numerazione anche adesso non è andato avanti più di così, ed a beneficio di chi si meravigliasse di tanta arretratezza ricorderò che i *tre* generi del greco antico (singolare, *duale* e plurale) altro non sono che il residuo di tale sistema di numerazione.
Ben presto il numero si rivelò un compagno utile sì, ma faticoso e difficile da maneggiare, e di fronte a questa situazione l'uomo dovette far ricorso al suo ingegno: difatti iniziò ad inventare una serie di attrezzi che erano intesi a rendergli più facile la vita. Fu usato un po' di tutto: dita, pietruzze, bastoncini, cordicelle, colorate e non, annodate e non. Ma il principe sovrano di *tutti* i calcoli antichi fu l'abaco, quello che ora viene detto "giapponese", ma che in realtà fu diffuso in tutto il mondo, sia pure con lievi varianti. Ad esso l'uomo fece ricorso per secoli, tanto che non è esagerato dire che esso fu il calcolatore del mondo antico. Forse un giorno parleremo anche dell'uso dell'abaco: per ora vi basti sapere che con esso si possono eseguire le quattro operazioni con velocità notevolissima. La pratica e l'incredibile agilità delle dita umane hanno modo di esplicarsi in questo funzionalissimo strumento con eccezionali risultati. Non è raro trovare in Giappone dei commessi che vi vendono calcolatori tascabili, e vi fanno il conto con un abaco!
Ecco quindi la risposta ad una famosa domanda: "Chissà come facevano i Romani a fare i calcoli?". Evidentemente non con il loro infernale sistema di numerazione, buono al massimo per indicare legioni o miglia stradali, ma con l'abaco, sul quale erano infilate delle pietruzze (dette "calculi" in latino!).
Abaci e tabelle furono a lungo gli unici sistemi di calcolo a disposizione dei matematici. Man mano che il sapere progrediva, divennero più complessi i conti da fare: ed infine, verso il 1400-1500, la situazione divenne disperata. Ancora una volta l'uomo si trovò di fronte al numero amico-nemico. E, così come il suo antenato aveva praticamente risolto con l'abaco i suoi problemi di aritmetica dei numeri interi, l'uomo rinascimentale inventò una serie di trucchi per dominare i numeri *non* interi.
Come fare (ed in fretta) una radice cubica? Come calcolare lati ed angoli di un triangolo? La principale responsabile di questa spinta fu la meccanica celeste, con le sue serie richieste di precisione, ma i prodotti che ne emersero in termini di progresso nei calcoli furono molto più importanti di quanto gli stessi ideatori avessero potuto supporre. A parte i prìncipi di questa famiglia di attrezzi (vale a dire i logaritmi ed

il calcolo grafico, con costruzioni a base di scale graduate, ancora oggi molto in uso), venne creata una serie di conoscenze sui metodi di approssimazione più rapidi ed efficienti, che oggi costituisce in ogni Istituto scientifico una parte non trascurabile della biblioteca, e che è stata ampiamente sfruttata nei moderni calcolatori: in fondo, se un metodo intelligente fa risparmiare tempo all'uomo, deve farlo risparmiare anche ad un calcolatore!
Gli inizi del calcolo *non* umano, ma affidato alle macchine, comparvero però solo nel 1600, e, dato lo stato della tecnologia dell'epoca, non ebbero poi molta fortuna. Nel secolo della Meccanica e dell'Astronomia ebbero senz'altro sorte migliore le scorciatoie di calcolo (come i logaritmi) che non le prime incredibili macchine per fare qualche somma o prodotto. Per confronto, tenete presente che i logaritmi (Napier, Briggs) sono nati a cavallo del 1600 e che le prime macchine calcolatrici nacquero nel 1642 (ad opera di Pascal) e nel 1671 (ad opera di Leibnitz). Il quale Leibnitz doveva poi veramente odiare gli aspetti manuali di questo lavoro se poteva scrivere: "È indegno di uomini dall'eccellente ingegno lo sprecare ore nella fatica del calcolare".

Le macchine dormirono quindi a lungo, e dovettero attendere lo sviluppo dell'industria, nel 1800, per uscire dal laboratorio dell'Inventore ed entrare nel mondo esterno. In compenso, la loro crescita fu esplosiva: in pratica, per poco più di 70 anni (fino al 1950 circa) la macchina calcolatrice, meccanica prima, elettromeccanica poi, dominò incontrastata. Ancora oggi troverete in molti negozi e banche le macchine "a manovella". Comunque, per confronto, sarà bene far notare che nel '60 una macchina elettromeccanica che faceva solo le 4 operazioni, radici quadrate ed accumulo di prodotti (niente numeri negativi, niente virgola automatica) impiegava circa *venti secondi* per fare *una* divisione, pesava una decina di chili e costava almeno un milione (di allora).
Dal 1936 il campo del calcolo meccanico ha fatto passi da gigante. Ancora una volta un progresso tecnologico ha reso possibile un progresso scientifico, e viceversa: le indagini sull'atomo (dell'inizio del secolo) hanno aperto le strade all'elettronica, e questa ha via via cominciato a far capolino nelle applicazioni pratiche. Timide dapprima, sempre più travolgenti poi. Del resto il calcolatore, sempre più complesso, versatile e veloce, ha reso possibili calcoli prima impensabili, e l'odierna fisica dello stato solido e la micro-elettronica devono la loro stessa esistenza a questo gioco di interazione teoria-strumento.
Ecco alcune tappe di questo sviluppo:

1936-41: Konrad Zuse (studente tedesco di ingegneria civile) costruisce lo Z3, a relais.

1943-46: la IBM costruisce a Cambridge (Mass.), nella Harvard University, il MARK-I, a relais anch'esso. Verrà smantellato nel 1959.

1943-46: Nella Pennsylvania University viene costruito ENIAC, il primo calcolatore veramente elettronico, tutto a valvole. 18000 valvole, 30 tonnellate, 200 KW di potenza assorbita. Pressappoco la potenza necessaria per muovere un treno medio alla velocità di 70 Km/h. Verrà usato fino al 1958. Nel solo 1952 vennero sostuite 19000 valvole.

1958: escono i primi calcolatori a transistors.

1963: escono i primi calcolatori a circuiti integrati.

1965: esce il primo "minicomputer" (PDP-8).

1967: tutti i calcolatori sono ormai a circuiti integrati.

1971: esce il primo "microprocessore" (INTEL).

1978-79: micro e mini computers hanno ormai un costo tale da permetterne l'acquisto a privati. Il calcolatore è divenuto veramente alla portata di tutti.

E per i grossi mostri? Si annunciano due vie nuove: La prima è perseguita da calcolatori supercompatti e superveloci (proprio perché supercompatti), i quali inoltre cercano, non appena possibile, di effettuare operazioni diverse contemporaneamente ("in parallelo") invece che una dopo l'altra ("in serie"). La seconda via è basata su un principio scoperto da tempo, ma non ancora sfruttato (l'effetto Josephson), che avviene a temperature molto basse (2-3 °K, cioè 270 °C sotto zero).
Il calcolatore raffreddato ad elio liquido sarà probabilmente il prossimo "grosso mostro". Siamo, come vedete, in piena fantascienza.

L'informazione nel calcolatore

È merito dei logico-matematici l'aver sviluppato la teoria dell'informazione, costruendo a poco a poco tutto un edificio matematico che, partito da pubblicazioni specialistiche e da lavagne universitarie, seguite da pochi ed assonnati studenti, ha poi trovato nella tecnologia moderna il suo vero sbocco. Ciò va detto a beneficio di quanti (magari anche fra di voi!) pensano che la "teoria" sia qualcosa che vale al più per gente che passa la sua vita chiusa in stanze polverose: in realtà la conoscenza è un procedimento complesso in cui la sperimentazione e la pratica si devono fondere con il momento di speculazione teorica. A questo proposito vi basti sapere che tutta (dico *tutta*) la moderna microelettronica è basata sulla meccanica quantistica, e per di più nei suoi aspetti più spinti e raffinati.
Fu Boole (*The Laws of Thought*, 1854) a porre i fondamenti della logica che porta il suo nome (logica "booleana"), e Shannon (1948) fu il primo a riconoscere l'importanza pratica del lavoro di Boole: per la prima volta infatti questo tipo di formalismo astratto venne applicato ad un problema concreto, cioè a delle reti telefoniche.

In cosa consiste dunque l'algebra booleana, e come mai ha tanta importanza nel moderno trattamento dell'informazione? La sua importanza sta nel riconoscimento del fatto che qualunque informazione, *di qualunque genere* (state bene attenti a questa frase!) può venire espressa come un'opportuna combinazione di entità fondamentali astratte, le quali possono assumere solo *due* valori. Poco importa come decidete di chiamare questi "valori": bianco e nero, su e giù, 0 e 1, vero e falso, aperto e chiuso ... scegliete voi.

Una cosa è importante: combinando queste entità voi potete trasmettere e manipolare *tutta* l'informazione che volete: parole, libri, musica, matematica... tutto è riconducibile a questo "atomo di informazione". Il quale poi si chiama ormai universalmente BIT, acrostico di *BI*nary Digi*T*, che però in inglese vuol dire anche "un pochino, un pezzettino, un bocconcino". È quindi un gioco di parole intraducibile, oltre ad essere - naturalmente - la testata di questa Gloriosa Rivista.
Nel calcolatore, quindi, ogni informazione è ridotta in bits. Nessun calcolatore al mondo è in grado di fare di più. In compenso, per questa sua intrinseca "stupidità" (dovuta al fatto che tutta l'informazione, di qualsivoglia natura, è ridotta nei suoi termini elementari), il calcolatore può eseguire le sue operazioni a velocità talmente incredibili da riuscire ad effettuare operazioni estremamente complesse in tempi assolutamente da primato.
Cominciamo quindi a vedere come si possono rappresentare *i numeri* all'interno del calcolatore. Per quanto l'elaborazione dei numeri sia l'uso più comune di un calcolatore, sarà bene che teniate presente che non è l'unico: *qualunque tipo di informazione può essere da esso elaborato,* ed in questo consiste la differenza fondamentale fra il calcolatore moderno e le calcolatrici dei tempi passati. Non è esagerato dire che il calcolatore estende le capacità del nostro cervello, così come la macchina ed il motore hanno esteso le capacità del nostro sistema muscolare.

Il sistema binario

Se dobbiamo rappresentare i numeri con dei bits, il sistema di rappresentazione più ovvio è il *sistema binario*, e, poiché molti di voi non lo conoscono, vediamo un po' di che si tratta. Noi siamo debitori agli Arabi, ed agli Indiani prima di loro, di una serie di idee nuove circa il modo di rappresentare i numeri:

- l'uso di simboli speciali per indicare i numeri (tutti gli altri popoli usavano le lettere dell'alfabeto)
- l'introduzione di un simbolo speciale per lo zero. Sia ZERO che CIFRA (= ZIFR) sono parole arabe!
- la notazione posizionale.

Sul primo e sul secondo punto non c'è molto da dire, salvo che il nostro sistema di numerazione usa dieci simboli diversi (cifre, e "digits" in inglese; "digitus" in latino vuol dire proprio "dito della mano"). Questa scelta deriva naturalmente dal fatto che noi abbiamo dieci dita.
Sul terzo punto invece dobbiamo riflettere: quando scriviamo 55, i due "5" hanno un significato diverso *a seconda della posizione che occupano.* Il primo cinque sta a simboleggiare un mucchio di cinquanta sassolini, il secondo un mucchio di soli cinque sassolini, e scrivere le due cifre di seguito sta ad indicare che il numero finale si otterrà sommando, cioé unendo, i due mucchi. Spostare una cifra di un posto a sinistra sta insomma ad indicare un numero *dieci volte* maggiore. In formule,

$$237 = 2 \cdot (10)^2 + 3 \cdot (10)^1 + 7 \cdot (10)^0$$

e così via. (Vi ricordo che $10^0 = 1$: se non ve lo ricordate andate a rivedervi il vostro libro di algebra!).
Ecco quindi come si possono indicare con le cifre del sistema decimale dei mucchietti di pietre: fino a nove

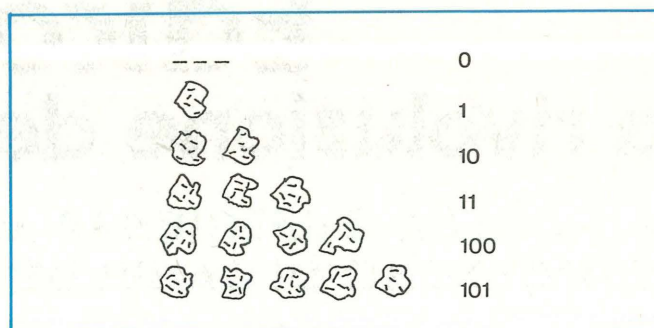



Figura 1 - Rappresentazione binaria di una quantità numerica.

pietre si usano le cifre che si hanno a disposizione. Aggiungendo un sasso non si hanno più cifre per indicare questo nuovo numero. Allora si usa un numero di *due* cifre (10), con la convenzione che l'"uno" spostato a sinistra stia ad indicare proprio il gruppo di *dieci* pietre, al quale è da aggiungere il numero indicato dalla seconda cifra (zero, cioè nulla, in questo caso).

Usando i bits non potremo più lavorare con dieci cifre, ma dovremo accontantarci di usarne due sole: useremo 0 e 1. In Figura 1 abbiamo un esempio di come si scrivono i numeri nel sistema binario, e penso che non avrete alcuna difficoltà a continuare a scrivere i numeri successivi.
Più in generale: le cifre del sistema binario sono solamente *due*, e quindi la notazione posizionale in questo sistema si basa su potenze successive di *due*. Quindi per esempio abbiamo

$$(11010)_2 = 1 \cdot (2)^4 + 1 \cdot (2)^3 + 0 \cdot (2)^2 + 1 \cdot (2)^1 + 0 \cdot (2)^0 = (26)_{10}$$

Il numeretto scritto in piccolo in basso a destra sta ad indicare la base in cui il numero viene scritto: quindi lo stesso numero ventisei si scrive 26 (o - per chiarezza - $(26)_{10}$) se si usa la notazione decimale, e 11010 se si usa la notazione binaria.
Se provate a scrivere alcuni numeri in notazione binaria, vi accorgerete ben presto del fatto che questi numeri si presentano come stringhe molto lunghe di uni e di zeri: evidentemente il sistema binario è molto poco pratico per gli usi normali: è difficile e lungo scrivere materialmente i numeri, e non parliamo di ricordarseli! Altro naturalmente è il caso del calcolatore, che non ha problemi di errori, di comodità e di facilità di lettura.
Abbiamo quindi già in pratica visto come si fa a passare dal sistema binario al sistema decimale: eccovi l'esempio precedente ripetuto in modo più esplicito:

$$(11010)_2 = 1 \cdot (2)^4 + 1 \cdot (2)^3 + 0 \cdot (2)^2 + 1 \cdot (2)^1 + 0 \cdot (2)^0 = \\ = 16 + 8 + 2 = \\ = (26)_{10}$$

Un pò più complicato è il passaggio inverso, ma eccovi una semplicissima ricetta, basata su successive divisioni per due, e con un formalismo che dovrebbe esservi familiare, visto che ricorda molto da vicino quello che avete imparato circa la scomposizione di un numero in fattori primi. Eccovi quindi le regole che dovete imparare:

- se il numero da cui partite è *pari*, scrivete 0 nella colonna dei resti e dividete il numero per due

Ediconsult
la rivoluzione economica

PREZZI IMBATTIBILI

CPU Microprocessore Z80A con: - 32K di memoria utente con 1 unità Floppy singola faccia-singola densità (tot. capacità 256K bytes) - un terminale video da 1920 caratteri - una stampante da 60 caratteri al secondo

Listino **L. 7.000.000** — Offerta Speciale **L. 6.150.000**

CPU microprocessore Z80A con: - 32K di memoria utente con 2 unità Floppy singola faccia-doppia densità (capacità 1 Megabytes) - un terminale video da 1920 caratteri - una stampante da 120 caratteri al secondo bidirezionale

Listino **L. 11.400.000** — Offerta Speciale **L. 10.000.000**

CPU microprocessore Z80A con: - 64K di memoria utente con 2 unità Floppy singola faccia-doppia densità (capacità 1 Megabytes) ed un disco fisso da 14,5 Megabytes o da 29 Megabytes - un terminale video da 1920 caratteri - una stampante da 120 caratteri al secondo bidirezionale

Listino **L. 19.500.000** — Offerta Speciale **L. 17.000.000**

CPU Microprocessore Z80A con: - 112K di memoria utente con 2 unità Floppy singola faccia-doppia densità per 2 utenti (capacità 1 Megabytes) - due terminali video da 1920 caratteri - una stampante da 120 caratteri al secondo bidirezionale

Listino **L. 15.600.000** — Offerta Speciale **L. 13.600.000**

SPECIALE

CPU Microprocessore 6502 oppure Z80 con: - 32K di memoria utente con 2 unità Floppy singola faccia-singola densità (capacità 512K Bytes) - un terminale video da 1920 caratteri - una stampante da 60 caratteri al secondo

Listino **L. 9.300.000** — Offerta Speciale **L. 8.150.000**

CPU Microprocessore 6502 oppure Z80 con: - 48K di memoria utente con 2 unità Floppy singola faccia-singola densità ed un disco da 74 Megabytes - un terminale video da 1920 caratteri - una stampante da 120 caratteri al secondo bidirezionale

Listino **L. 28.000.000** — Offerta Speciale **L. 24.000.000**

Personal Computer con microprocessore 6502 - 8K Basic in ROM e 4K di memoria RAM - registratore di cassetta e video monitor

Listino **L. 1.200.000** — Offerta Speciale **L. 1.050.000**

CPU Microprocessore Z80A con: - 32K di memoria utente con 2 unità a mini-floppy singola faccia-doppia densità (tot. capacità 360K Bytes) un video da 1920 caratteri ed una stampante da 60 caratteri al secondo

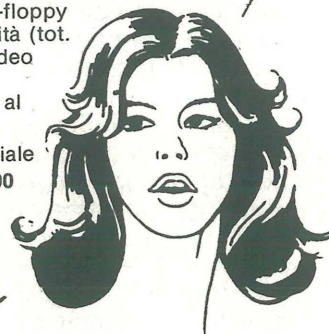
Listino **L. 6.700.000** — Offerta Speciale **L. 5.800.000**

CPU Microprocessore Z80A con: - 32K di memoria utente con 2 unità mini-floppy doppia faccia-doppia densità (tot. capacità 720K bytes) un video da 1920 caratteri ed una stampante da 120 caratteri al secondo bidirezionale

Listino **L. 9.100.000** — Offerta Speciale **L. 7.900.000**

Terminale Video Soroc 24 linee x 80 colonne

Listino **L. 1.350.000**

Offerta Speciale **L. 1.200.000**

QUESTA OFFERTA È VALIDA PER IL PERIODO 15 FEBBRAIO-15 APRILE 1980

EDICONSULT

MONZA - Via Rosmini 3, Tel. 039/389.850 - 360.727
ROMA - Via Busiri Vici 5, Tel. 06/5809392

TELEFONATE,
VEDIAMOCI,
MA NON PERDETE L'OCCASIONE

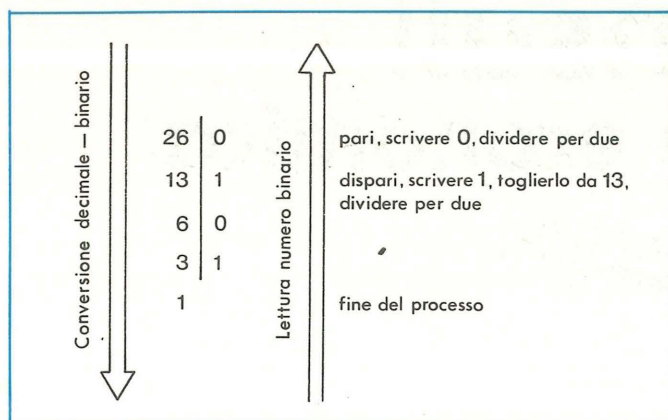



Figura 2 - Conversione dal sistema decimale al sistema binario.

- se il numero da cui partite è *dispari*, scrivete 1 nella colonna dei resti, *sottraete* uno dal numero di partenza e poi dividete per due.

Un esempio (sempre sul numero 26, per semplicità) è riportato in Figura 2.

A questo punto il numero si legge nel senso della freccia, ed è -ovviamente- 11010.
Vi consiglio di esercitarvi a convertire un po' di numeri binari in notazione decimale e viceversa: è un utilissimo esercizio per tutto ciò che verrà poi.

Il byte e le parole

All'interno del calcolatore l'informazione, ridotta in bits, è organizzata però sotto forma di *gruppi* di bits: anche in questo caso non c'è niente di concettualmente nuovo. Infatti anche in una biblioteca l'informazione è si ridotta alle lettere, ma queste sono poi organizzate in strutture più complicate, quali parole, righe, pagine, volumi. Nel calcolatore questi gruppi di bits hanno una caratteristica: sono *indirizzabili*. Anche se vedremo più avanti ciò che questa parola vuole esattamente dire, basterà per ora notare che ognuno di questi gruppi può essere univocamente individuato entro il calcolatore, in modo analogo a ciò che accade alle pagine di un libro.
Quanti sono i bits che costituiscono uno di questi gruppi? Be', qui la tecnologia ha proceduto per tentativi ed errori, ma alla fine pare che un compromesso ragionevole e più o meno universalmente accettato sia stato raggiunto con gruppi di *otto bits*. Ognuno di questi gruppi si chiama anche BYTE, con il solito intraducibile gioco di parole: infatti "bit" significa anche "bocconcino", e "bite" significa "morso, boccone". "Byte" invece non significa nulla, è un neologismo, che però si pronuncia allo stesso modo di "bite".
Un byte è quindi la minima quantità di informazione che si può estrarre dal calcolatore in una singola operazione, o cui comunque ci si può riferire, e ciò - salvo rarissime eccezioni- vale per tutti i calcolatori di oggigiorno.
Così pure si misura la capacità di memoria di un calcolatore indicando quanti bytes vi possono trovare posto. Anzi, dato che questi numeri sono in genere molto elevati, li si indica di solito coi loro mulripli: migliaia (kilo-bytes o K-bytes) e milioni (mega-bytes o M-bytes). Così un calcolatore "da 32 K" sta ad indicare un calcolatore nella cui memoria possono trovare posto *circa* 32000 bytes (32 kilo-bytes). Vi chiederete il perché di questo strano "circa": in realtà il numero esatto è 32768 bytes, cioè 2^{15} bytes. Per comodità si trascura quel 768, ed invece di dire che il calcolatore è da 32768 bytes, si dice per brevità che è "da 32 K". In fondo basta capirsi ...
Normalmente quindi, così come il bit rappresenta l'unità elementare di informazione, il byte rappresenta il più usato (ormai universalmente) raggruppamento di bits. Ciò non toglie che anche i bytes non possano essere raggruppati insieme, a due a due, o a quattro a quattro, o in altro modo. Uno di questi raggruppamenti prende in genere il nome di *parola*. Tipicamente mini-computers ed in nuovi microcomputers usano parole da 16 bits (quindi da *due* bytes). I grossi bestioni hanno regole loro: 32 bits (4 bytes) è la scelta IBM, mentre 48 bits (6 bytes) è la scelta CDC nel campo del calcolo scientifico. Noi però ci limiteremo in pratica a considerare solo il caso dei 16 bits, che è poi quello più comune.

Rappresentazione esadecimale

Anzitutto vediamo come si può scrivere un byte: la ricetta sarebbe ovvia da quanto già vi ho detto: basterebbe scrivere uno di seguito all'altro otto bits. Per esempio così:

1 0 1 1 0 1 0 1

ma questa notazione ha il difetto di essere lunga e di favorire gli sbagli in modo straordinario. Conviene quindi scegliere un nuovo sistema di numerazione: stavolta si sceglie un sistema in base *sedici* (e per ciò detto ESADECIMALE). Qui si presenta però una nuova difficoltà: poiché le cifre che noi abbiamo a disposizione sono solamente dieci, per far funzionare

Tabella I -

Numero	Sistema decimale	Sistema binario	Sistema esadecimale
zero	0	0	0
uno	1	1	1
due	2	10	2
tre	3	11	3
quattro	4	100	4
cinque	5	101	5
sei	6	110	6
sette	7	111	7
otto	8	1000	8
nove	9	1001	9
dieci	10	1010	A
undici	11	1011	B
dodici	12	1100	C
tredici	13	1101	D
quattordici	14	1110	E
quindici	15	1111	F
sedici	16	10000	10
diciassette	17	10001	11
diciotto	18	10010	12
diciannove	19	10011	13
venti	20	10100	14
ventuno	21	10101	15
vendidue	22	10110	16
ventitre	23	10111	17
ventiquattro	24	11000	18
venticinque	25	11001	19
ventisei	26	11010	1A
ventisette	27	11011	1B

il meccanismo di una notazione posizionale in base sedici dobbiamo inventare altre sei cifre. Con scarso senso di fantasia -devo dire- si è deciso di usare le prime sei lettere dell'alfabeto. E, poichè penso che un esempio valga mille discorsi, vi rimando alla Tabella I, nella quale sono scritti un po' di numeri utilizzando i vari sistemi di notazione. Esaminando tale tabella vi accorgerete che un gruppo di 4 bits può essere espresso da una sola cifra esadecimale (non per nulla $2^4 = 16$!). E quindi un byte può essere espresso con *due* cifre esadecimali.
Notate che, in base ai dettami della notazione posizionale, scrivere 2A in esadecimale vuol dire scrivere

$$2 \cdot (16)^1 + (A)_{16} = 2 \cdot 16 + 10 = 32 + 10 = 42$$

La semplificazione dunque sta in questo, che ogni byte si può scrivere come un numero di sole *due* cifre esadecimali. Il massimo numero inoltre che possiamo scrivere con due cifre esadecimali è quindi FF, cioè

$$(FF)_{16} = (15)_{10} \cdot (16)^1 + (15)_{10} = 240 + 15 = 255$$

e, tenendo conto anche dello zero, un byte può esprimere quindi 256 informazioni diverse.

Un esercizio

Provate ora a fare un po' di pratica.

I) Scrivete in binario e in decimale i seguenti numeri esadecimali:

1F, CA, FE, E2, 7C, 77, 99

II) Convertite in esadecimale i seguenti numeri decimali:

7, 34, 92, 137; 232, 74, 143, 32

Potete inventare un metodo, analogo a quello che vi ho spiegato, per effettuare questo tipo di operazione?

III) Provate (basandovi su ciò che sapete delle somme col sistema decimale) a definire le regole per sommare insieme numeri binari ed esadecimali. Per esempio

```
101101 +      1AB32 +
 10110 =       724F=
-------      -------
.......      .......
```

Le risposte al prossimo numero.

Cos'è un programma

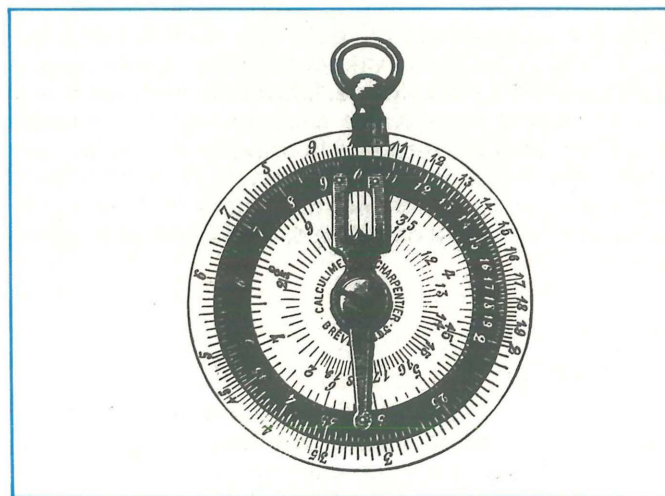
E veniamo alla seconda parte di questo articolo. Facciamo quindi improvvisamente un salto in avanti, e supponiamo già di sapere come funziona un calcolatore. Basterà tener presente a questo proposito che le parole all'interno di un calcolatore possono venir interpretate in modi diversi: o come informazioni da maneggiare (*dati* cioè) o come azioni da compiere (dette anche *istruzioni*). In entrambi i casi si tratta sempre di parole, espresse in codice binario. È compito della circuiteria (e vedremo poi come) decidere quale istruzione deve venire eseguita: l'insieme di tutte le istruzioni e di tutti i dati prende il nome di *programma.*

Se quindi dovessimo scrivere un programma, basterebbe che scrivessimo tutte le parole che lo compongono (in codice binario) e che le introducessimo nel calcolatore. In effetti una volta si faceva proprio così e lo si fa ancora adesso con i piccoli calcolatori, ma capirete bene che, quando un programma divenisse di sole 10000 parole, sarebbe sconsigliabile il solo scriverlo in codice binario (ricordatevi che ogni parola è composta da 16 bits!).
Sono stati quindi sviluppati dei trucchi opportuni per aggirare l'ostacolo. Essenzialmente si tratta di questo: invece di scrivere le parole di istruzione ed i dati in codice binario, scriviamoli in lettere, numeri e simboli. Poi scriviamo un programma che *traduca* tutto ciò in modo comprensibile al calcolatore.
Questo tipo di programma andrà scritto una volta per tutte, sarà magari molto elaborato, ma una volta scritto e funzionante permetterà di stendere dei programmi in modo semplice, lasciando al calcolatore il compito difficile, lungo e noioso, di tradurseli in modo comprensibile (a lui).
Questi tipi di programmi si chiamano *assemblatori*, *compilatori* ed *interpretatori*, mentre il complesso di regole che permette di scrivere un programma si chiama anche *linguaggio*.

Assemblatori

Un assemblatore (assembler) lavora su programmi scritti nel più semplice linguaggio possibile, molto vicino allo stesso linguaggio del calcolatore: ogni istruzione eseguita dal calcolatore viene indicata e descritta, anche se in modo più o meno descrittivo ed intuitivo.
Così la somma si indica di solito con la parola ADD, mentre un confronto (in inglese "to compare") si indica magari con COM o con CMP. Scrivere programmi in questo linguaggio è in genere impresa molto lunga e complessa, anche se è proprio con questo linguaggio che si sfruttano a pieno le possibilità del calcolatore.

Compilatori

Se però doveste avere problemi appena appena un po' complessi, scrivere programmi con questo linguaggio diverrebbe un autentico supplizio. È proprio in funzione dei problemi che emergono nel calcolo

Regolo calcolatore fi forma circolare di Charpentier, 1870.

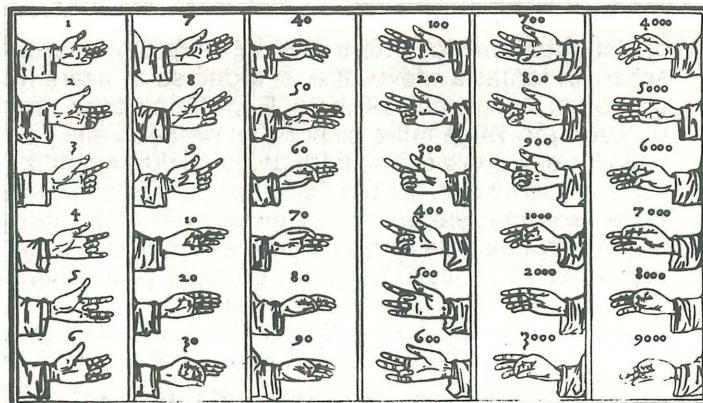
scientifico che fin dal 1958 sono stati sviluppati dei linguaggi più complessi, simili al linguaggio matematico, che permettono di scrivere formule complesse in modo semplice e compatto. L'ALGOL e soprattutto il FORTRAN sono -fino ad oggi- gli incontrastati dominatori del campo, anche se si affaccia ora all'orizzonte il più moderno ed evoluto PASCAL, che spero abbia molta fortuna, come linguaggio del futuro. Questi linguaggi permettono di scrivere delle operazioni in modo semplice per l'uomo, ma lasciano al calcolatore un compito molto gravoso: non solo infatti deve tradurre il tutto nel suo linguaggio, ma deve anche cercare tutti i trucchi possibili affinché il programma sia il più breve e veloce possibile. Il calcolatore non inventa nulla, beninteso, ma spetta anzi a colui che costruisce il programma di traduzione di "pensarle tutte". Non a caso questo tipo di programmi, detti *compilatori* (compilers), richiedono il lavoro di molti ed esperti programmatori.

Dopo che un assemblatore o un compilatore ha lavorato, il risultato è un programma nel linguaggio del calcolatore (detto anche *linguaggio macchina*) il quale però in genere non può essere ancora eseguito: di solito occorre un'opera di cucitura di pezzi, di aggancio a programmi standard del calcolatore, etc, la quale prende il nome di Link-Edit ("aggancia e scrivi in bella copia") o di Task-Building ("costruzione del programma") e che è affidata ad un *secondo* programma.

Solo dopo queste "due passate", il calcolatore è in possesso di un programma detto anche LOAD-MODULE (modulo caricabile) o TASK (compito, lavoro) che può essere anche conservato, magari su nastro o disco, e di là direttamente *caricato* (loaded) in memoria. Ciò che esce invece da un assemblatore o da un compilatore si chiama OBJECT MODULE (modulo oggetto), mentre il programma di partenza si chiama SOURCE MODULE (modulo sorgente).

Supponiamo dunque che abbiate scritto il vostro programma, e che attraverso queste due passate (compilazione e link-edit) abbiate costruito il load-module, il programma che il calcolatore capisce e può eseguire. Introducete i dati, fate partire il programma, e vi accorgete che il programma non funziona. Cosa è successo? Evidentemente avete commesso un errore. Allora riprendete in mano il source-module, lo esaminate, trovate un errore, lo correggete, rifate le due passate, riprovate e ... trovate che il programma non funziona ancora! E così andate avanti in questa fase di spulcio (o di DEBUG = "togliere il bug", cioè la cimice), finché tutto funziona correttamente. Fase lunga, laboriosa, sofferta, che in genere porta le persone sull'orlo dell'esaurimento nervoso. Tanto è vero che la bravura di un programmatore e la sua esperienza non si giudicano tanto dalla sua abilità di scrivere dei programmi intelligenti, quanto dalla sua capacità di cacciare e correggere gli errori!

Interpretatori

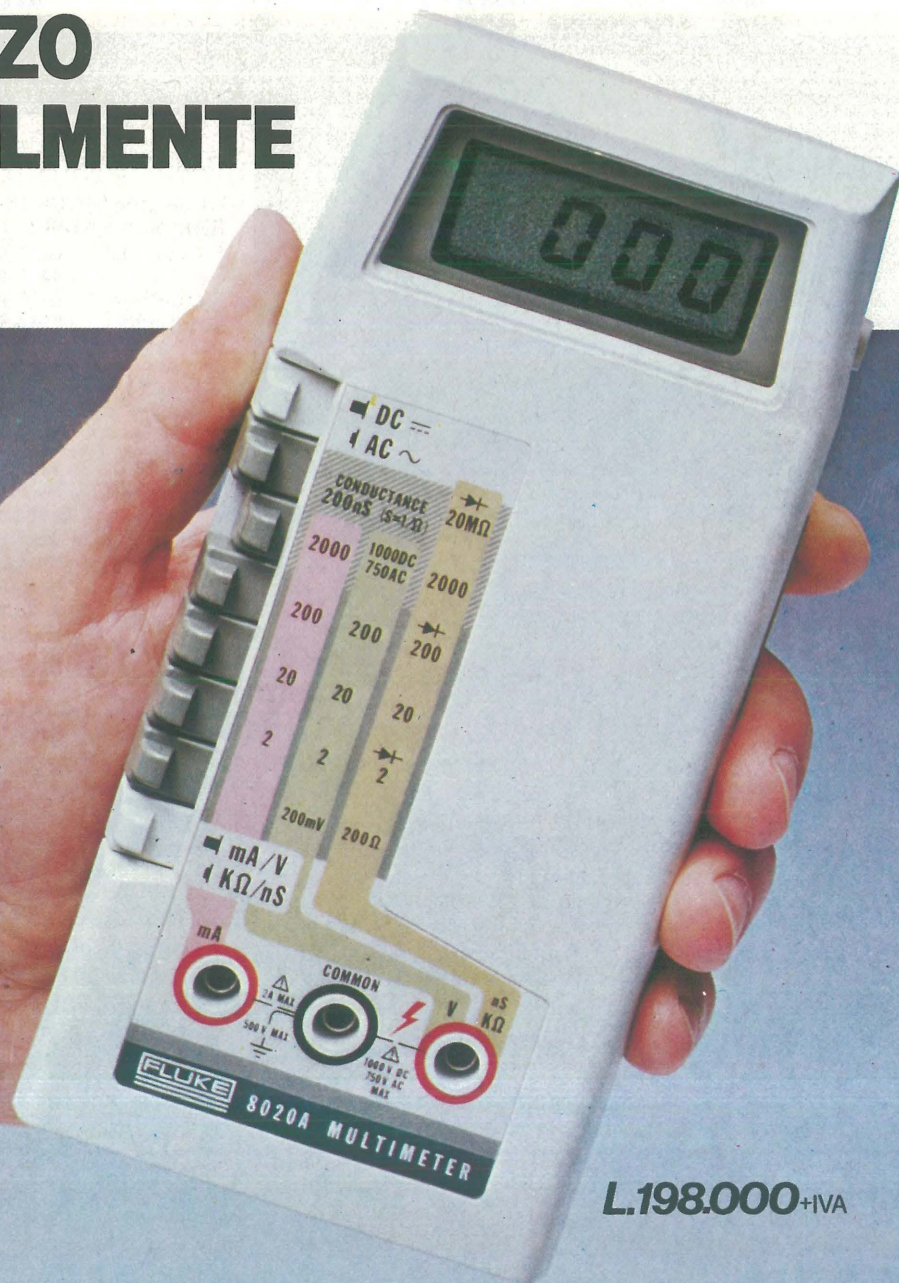
Nei calcolatori micro, ed in quelli da tavolo, è quindi stato introdotto un nuovo linguaggio (il BASIC appunto) che cerca di evitare al massimo questo tipo di pene all'utente, anche a costo di sacrificare qualcosa soprattutto in termini di lunghezza e di rapidità di esecuzione. Non è però un grandissimo sacrificio: infatti la memoria dei calcolatori odierni è molto a buon mercato e la loro velocità è incredibilmente alta. Quindi in fondo la filosofia non è sbagliata: se il programma finale occupa più memoria ed impiega più tempo, ed a questo prezzo si evitano all'uomo dei potenti mal di testa, ... be', tutto di guadagnato!

Il metodo più sofisticato di calcolo digitale è dovuto a Luca Pacioli che nel 1594 pubblicò un'opera: "Summa de Arithmetica", da cui è tratta la tavola, che raffigura il modo di rappresentare con le dita i numeri da 1 a 9999.

In cosa dunque differisce la filosofia del BASIC da quelle che abbiamo visto finora? Anzitutto è un linguaggio ad alto livello, come il FORTRAN (cui peraltro è molto simile) ed ha inoltre una particolarità: *ogni* sua istruzione deve essere numerata. Il programma di traduzione (che è detto *interpretatore* e non *compilatore*) traduce in linguaggio macchina solo una istruzione BASIC per volta, e non tutto il programma. Va detto -a scanso di equivoci- che una istruzione BASIC può essere molto complessa, e non è certo confrontabile con una singola istruzione del calcolatore. Comunque, la traduzione in linguaggio macchina viene fatta dall'interpretatore immediatamente prima di eseguire l'istruzione medesima.

Quindi in sostanza un programma scritto in BASIC viene capito dalla macchina come una serie di programmi (uno per ogni istruzione) che devono venire tradotti in linguaggio macchina ed eseguiti, uno di seguito all'altro: ciò che si chiama il *run* (la corsa, o esecuzione) di un programma in BASIC, si configura come una serie di cicli traduzione-esecuzione.

Va da sé che il programmatore ci guadagna enormemente nella fase di debug: infatti modificare, togliere, aggiungere istruzioni è semplicissimo, e vedere l'effetto che le modifiche hanno è altrettanto semplice: basta far ripartire il programma BASIC, senza dover passare ogni volta attraverso le "due passate" di compilazione e link-edit che avevamo menzionato prima: è in questo senso che il linguaggio BASIC viene detto "interattivo".

Per questa prima panoramica penso che basti: è già abbastanza se avete chiaro il concetto di ciò che succede quando si esegue un programma in BASIC, e se vi è altrettanto chiaro che ogni istruzione BASIC deve essere numerata. Nelle prossime puntate entreremo subito nei dettagli del linguaggio, e vedremo come si possono scrivere dei programmi in BASIC, e come si possono correggere. ■

IL MULTIMETRO PROFESSIONALE*
DAL PREZZO
INCREDIBILMENTE
BASSO
*della FLUKE
naturalmente!
DC
AC
CONDUCTANCE 200nS
20MΩ
2000
1000DC 750AC
200mV
200Ω
mA/V
KΩ/nS
mA
COMMON
V
nS KΩ
FLUKE
8020A MULTIMETER
L.198.000+IVA
Se non hai ancora acquistato un Fluke mod. 8020 probabilmente è solo perché non sai che:
● Ha 3 digit e mezzo (± 1999) ● E' protetto anche nella scala degli Ohm fino a 500 V ● E' protetto contro impulsi di tensione fino a 6000 V ● Può misurare 6 funzioni con 26 scale - (Vcc, Vca, mAcc, mAca, Ω, S) ● Può misurare resistenze fino a 10.000 Megaohm (sì, non è un errore di stampa) usando la funzione conduttanza con lettura in Siemens (S = 1/OHM) ● Può misurare diodi e resistenze senza dissaldarli dai circuiti ● Può misurare il coefficiente β dei transistori □ Può sopportare urti e vibrazioni secondo le norme MIL-T-28800
SISTREL
SOCIETÀ ITALIANA STRUMENTI ELETTRONICI S.p.A.
Via Pelizza da Volpedo 59 - 20092 CINISELLO B.
tel. (02) 6181893 - Telex 320346
Via Giuseppe Armellini 39, 00143 ROMA. Tel. (06) 5915553 - Telex 680356
Via Cintia Parco S. Paolo 35, 80126 NAPOLI - Tel. (081) 7679700
□ Ricevere un'offerta □ La visita di un Vs. Tecnico □ Essere inseriti nel Vs. mailing list.
NOME
COGNOME
VIA
TEL.
CAP.
CITTA'
DITTA
REPARTO
mod. 8020 A

SALOTA
CARATTERISTICHE TECNICHE MFC 512 SALOTA
Unità centrale da 16 a 64KB di RAM per mono utenza
RAM fino a 512KB in sistema multiterminale.
Floppy disk: da 1 a 3 mini (5.25")
da 1 a 4 standard singola o doppia faccia
Hard disk: da 5 a 300 MB.
Linguaggi: CBASIC
COBOL
FORTRAN IV
MACRO ASSEMBLER
PASCAL
BASIC MULTI USER
Terminali: Video con tastiera
Stampanti: da 80 a 132 colonne da 80 a 250 caratteri al secondo
Software: pacchetti standard per gestione magazzino, fatturazione, contabilità, calcolo scientifico, calcolo finanziario, didattica.
Personalizzazioni ed esigenze particolari in brevissimo tempo.
PLAE
PRODOTTI
LOMBARDI
AUTOMAZIONE
ELETTRONICA
Via Curtatone, 16 - 20098 S. GIULIANO MILANESE
Tel. 02 - 9880147 (MI)
SALOTA

Dama cinese

di **Gloriano Rossi** *(I 2 KH)*

Questi orientali devono senz'altro avere una mente estremamente contorta.

Fra tutti i giochi che sono riusciti ad inventare nel corso della loro millenaria storia non ne ho trovato alcuno che avesse la caratteristica di essere al tempo stesso complicato ed idiota. La dama cinese è uno di quei giochi che eccellono per semplicità, ma la tempo stesso difficile nella soluzione ed appassionante.

Regole del gioco

La scacchiera all'inizio del gioco è formata da 32 palline nere e da 33 caselline. Una sola quindi, alla partenza, è considerata casella libera.

Scopo del gioco è quello di rimanere con una sola pallina nera. Per poter far ciò occorre riuscire a mangiare tutte le altre palline secondo un limitatissimo numero di regole. Ogni mossa per essere eseguita deve portare come effetto una mangiata di una sola pedina. Le mosse potranno essere eseguite solamente in senso orizzontale e verticale, quindi non si può muovere in obliquo. Una sola pedina per volta può essere mangiata. La fine del gioco è determinata dal fatto di non poter più eseguire mosse "mangerecce".

La scacchiera si presenta come quella di Figura 1; ogni casella è contrassegnata da un determinato numero che per ragioni di programma ha un valore che può risultare incomprensibile.

A titolo di esempio dirò che la prima mossa che posso effettuare è quella di spostare la pedina che si trova nella casella n. 59 e metterla nella casella libera n. 41: avrò come effetto il mangiare la pedina sita in 50 e il liberare la casella 50 e 59. E così via fino al tentativo di rimanere con una sola pedina nera.

Il listing o listato (v. Figura 2), che dir si voglia, è già predisposto per uno dei più classici dei personal computer: il Pet 2001 della Commodore.

Per i possessori del Pet apparirà sullo schermo una scacchiera fissa grande con i numeri di riferimento delle caselline come quella di Figura 1, quindi in piccolo in basso a sinistra la scacchiera in evoluzione, mossa per mossa.

Per chi non fosse possessore di questo tipo di computer dirò che tutte le istruzioni Basic di questo programma sono standard, tranne che per quei caratteri scritti in negativo le cui funzioni sono:

- carattere S in negativo = ritorno nella posizione base dello schermo, cioè in alto a sinistra (Home)
- carattere Q in negativo = salto riga (freccia verso il basso), equivalente ad una normale istruzione di PRINT senza nessun contenuto

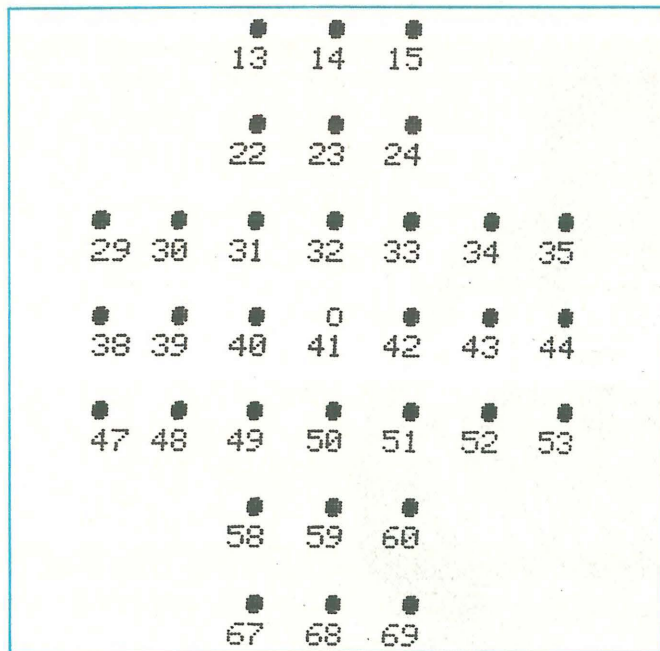



Figura 1 - Esempio della scacchiera al momento della partenza del gioco, notare la numerazione delle caselline. Tali numeri sono gli unici riferimenti frà il programma ed il giocatore.

- segno del cuore in negativo = pulisci lo schermo e ritorno alla posizione base (Clear)

Tutte le altre istruzioni devono essere uguali per tutti.

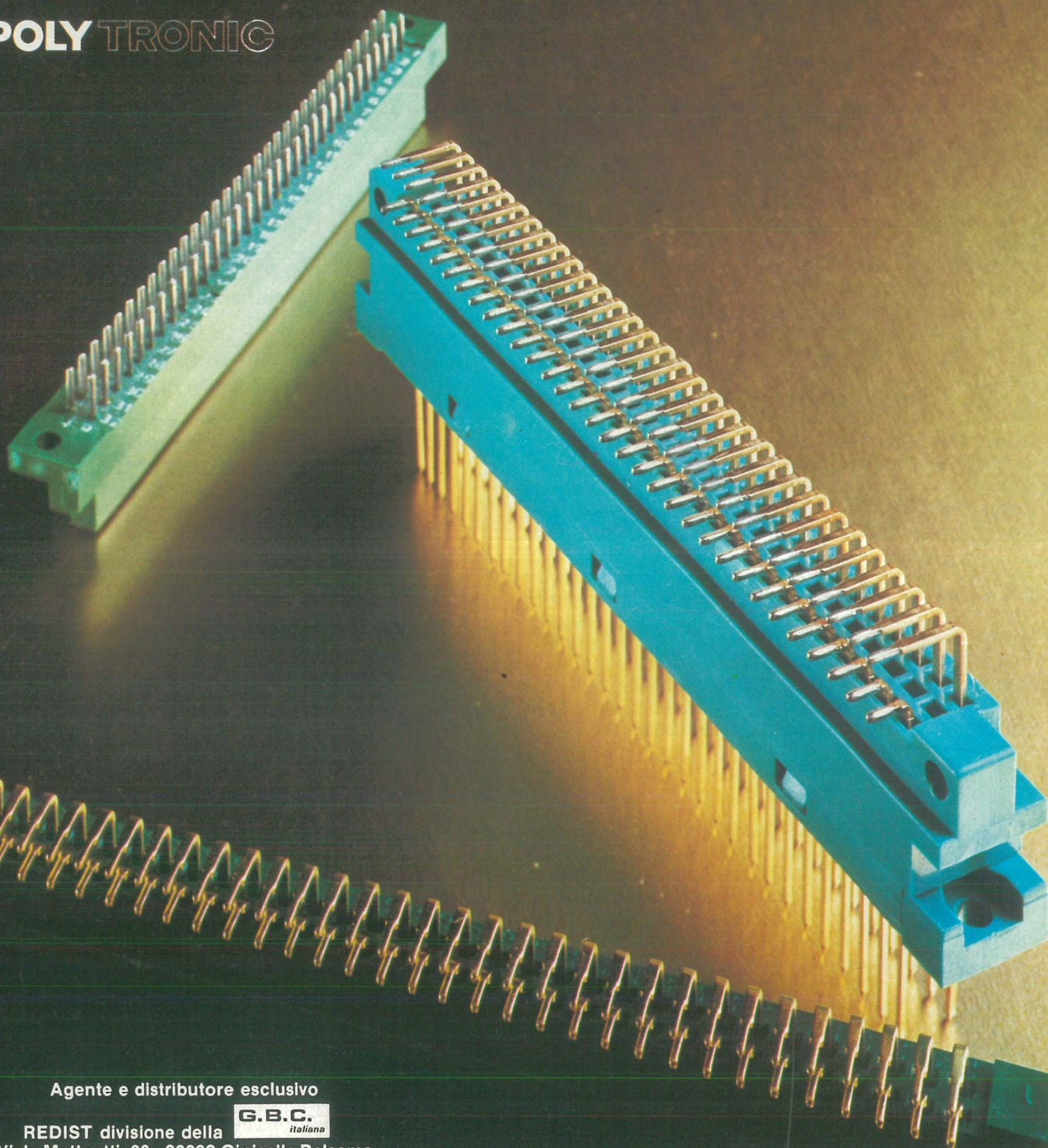
REMarks

Riga 4 - nella prima istruzione del programma si definisce il dimensionamento delle tre tabelle; le prime due di carattere numerico, l'ultima alfanumerico. La tabella B avrà 70 elementi utilizzati, quella definita con la lettera T 81 elementi numerici. Il dimensionamento in memoria definito con A$ conterrà le 14 stringhe alfanumeriche che verranno poi definite nelle istruzioni delle *righe 5 ... 18*. Il comando di *riga 20*, quello di PRINT del cuoricino in negativo, ci permette di pulire lo schermo video da ogni contenuto preesistente. Per visualizzare la scacchiera guida sono sufficienti le *righe 22, 23 e 24*, che tramite il contenuto della tabella A$ eseguito l'Edit (stampa) che ci interessa.

I valori di base della tabella T vengono introdotti durante l'esecuzione delle istruzioni *29 ... 50*. Come ho avuto modo di dire all'inizio della partita esiste una sola casella libera, quella centrale contrassegnata con il numero 41, che viene inizializzata durante l'esecuzione della *riga 65*. Dalla *riga 75* alla *83* si inizializza la tabellina B. Anche per questa tabella si forza il corrispondente valore della casella vuota di inizio partita.

A questo punto abbiamo finito la preparazione del "campo di gioco" e le istruzioni che seguono si riferiscono alla partita vera e propria. *In riga 100* si esegue il comando di ritorno alla posizione base sullo schermo quindi un riempimento della prima riga a spazio al fine di pulire detta riga dal contenuto precedente. La *riga 105* contiene l'istruzione di ritorno alla posizione base e quindi si esegue la richiesta del numero corrispondente della pallina che si vuole spo-

```
PROGRAMMA DAMA CINESE

 4 DIMB(70),T(9,9),A$(14)
 5 A$(1)="        ●   ●   ●"
 6 A$(2)="       13  14  15"
 7 A$(3)="        ●   ●   ●"
 8 A$(4)="       22  23  24"
 9 A$(5)="●   ●   ●   ●   ●   ●   ●"
 10 A$(6)="29 30  31  32  33  34  35"
 11 A$(7)="●   ●   ●   o   ●   ●   ●"
 12 A$(8)="38 39  40  41  42  43  44"
 13 A$(9)="●   ●   ●   ●   ●   ●   ●"
 14 A$(10)="47 48  49  50  51  52  53"
 15 A$(11)="        ●   ●   ●"
 16 A$(12)="       58  59  60"
 17 A$(13)="        ●   ●   ●"
 18 A$(14)="       67  68  69"
 20 PRINT"♥"
 22 FORI=1TO14STEP1:PRINTTAB(14)A$(I):
 23 IFI=2ORI=4ORI=6ORI=8ORI=10ORI=12THENPRINT
 24 NEXTI
 28 REM ** INIZIO GIOCO **
 29 FORR=1TO9:FORC=1TO9
 31 IF(R-4)*(R-5)*(R-6)=0THEN40
 32 IF(C-4)*(C-5)*(C-6)=0THEN40
 35 T(R,C)=-5
 36 GOTO50
 40 IF(R-1)*(C-1)*(R-9)=0THEN35
 42 T(R,C)=5
 50 NEXTC:NEXTR
 65 T(5,5)=0:GOSUB500
 75 FORW=1TO33:READM
 79 DATA13,14,15,22,23,24,29,30,31,32,33,34,35,38,39,40,41
 81 DATA42,43,44,47,48,49,50,51,52,53,58,59,60,67,68,69
 83 B(M)=-7:NEXTW
 86 B(41)=-3
 100 PRINT"■                    ":
 105 INPUT"■ FAI LA TUA MOSSA ";Z
 110 PRINT"■■■■                 ":
 115 IFB(Z)=-7THEN140
 120 PRINT"■■■■MOSSA ERRATA":GOTO100
 140 INPUT"■■■■ DOVE ";P
 150 IFB(P)=0THEN120
 153 IFB(P)=-7THEN120
 156 IFZ=PTHEN100
 160 IF((Z+P)/2)=INT((Z+P)/2)THEN180
 170 GOTO120
 180 IF(ABS(Z-P)-2)*(ABS(Z-P)-18)<>0THEN120
 190 GOSUB1000:GOSUB500:GOSUB1500:GOTO100
 500 REM   PRINT
 501 PRINT"■■■■■■■■■■■■■■■■"
 510 FORX=1TO9:FORY=1TO9
 525 IF(X-1)*(X-9)*(Y-1)*(Y-9)=0THEN550
 530 IF(X-4)*(X-5)*(X-6)=0THEN570
 540 IF(Y-4)*(Y-5)*(Y-6)=0THEN570
 550 GOTO610
 570 IFT(X,Y)<>5THEN600
 580 PRINTTAB(Y);"●";
 590 GOTO610
 600 PRINTTAB(Y);"o";
 610 NEXTY:PRINT:NEXTX:RETURN
 1000 REM AGGIORNAMENTO
 1005 C=1:FORX=1TO9
 1020 FORY=1TO9
 1030 IFC<>ZTHEN1220
 1040 IFC+2<>PTHEN1080
 1045 IFT(X,Y+1)=0THEN120
 1050 T(X,Y+2)=5
 1060 T(X,Y+1)=0:B(C+1)=-3
 1070 GOTO1200
 1080 IFC+18<>PTHEN1130
 1085 IFT(X+1,Y)=0THEN120
 1090 T(X+2,Y)=5:T(X+1,Y)=0:B(C+9)=-3
 1120 GOTO1200
 1130 IFC-2<>PTHEN1170
 1135 IFT(X,Y-1)=0THEN120
 1140 T(X,Y-2)=5:T(X,Y-1)=0:B(C-1)=-3
 1160 GOTO1200
 1170 IFC-18<>PTHEN1220
 1175 IFT(X-1,Y)=0THEN120
 1180 T(X-2,Y)=5:T(X-1,Y)=0:B(C-9)=-3
 1200 B(Z)=-3:B(P)=-7
 1210 T(X,Y)=0:GOTO1240
 1220 C=C+1
 1225 NEXTY
 1230 NEXTX
 1240 RETURN
 1500 REM   CHECK FINE GIOCO
 1505 F=0:FORR=2TO8:FORC=2TO8
 1530 IFT(R,C)<>5THEN1580
 1535 F=F+1
 1540 FORA=R-1TOR+1
 1545 T=0
 1550 FORB=C-1TOC+1
 1560 T=T+T(A,B)
 1561 NEXTB
 1564 IFT<>10THEN1567
 1565 IFT(A,C)<>0THEN1630
 1567 NEXTA
 1568 FORX=C-1TOC+1
 1569 T=0
 1570 FORY=R-1TOR+1
 1571 T=T+T(Y,X)
 1572 NEXTY
 1573 IFT<>10THEN1575
 1574 IFT(R,X)<>0THEN1630
 1575 NEXTX
 1580 NEXTC
 1590 NEXTR
 1600 REM  FINE DEL GIOCO
 1605 PRINT"♥■■■   IL  GIOCO  E'  FINITO"
 1610 PRINT"■■■RIMANGONO ";F;" PEZZI"
 1611 IFF<>1THEN1615
 1612 PRINT"■■■ECCEZIONALE BRAVISSIMO"
 1615 PRINT"■■■■■VUOI GIOCARE ANCORA ?";:INPUTA$
 1617 IFLEFT$(A$,1)="N"THEN2000
 1618 RESTORE:RUN
 1620 STOP
 1630 RETURN
 2000 PRINT"■■■■   C I A O "
 2010 END
```

Figura 2 - Listing del programma in Basic.

stare. In *riga 110:* ritorno, salto tre righe e pulizia della riga corrispondente dal contenuto precedente. Tramite la tabella B con indice uguale al valore impostato si controlla (*riga 115*) che effettivamente esista una pallina nera nella posizione indicata[B(Z)]. Se il sopradetto contenuto risultasse uguale al valore negativo di 7 si passa direttamente alla richiesta di quale casella dovrà ospitare la pallina in esame, altrimenti in caso contrario si stamperà un messaggio di errore (*riga 120*) e si ritornerà alla domanda primaria.

Il contenuto della tabellina B indicizzata con il valore della destinazione P dovrà avere un valore corrispondente a –3 altrimenti il programa rimanderà sempre alla richiesta di base. Ritengo abbastanza interessante la soluzione di controllo di validità delle mosse. Questo tipo di controllo viene eseguito con le istruzioni di *righe 160 ... 180* che ci permetteranno la verifica di un eventuale tentativo di eseguire un salto superiore alle due caselle, oppure se detto salto fosse il risultato di uno spostamento obliquo.

In *riga 190* si richiamano tre subroutines: aggiornamento, stampa ed infine controllo di fine gioco, quindi se l'esecuzione non fosse già uscita durante lo svolgimento di queste tre funzioni il programma torna alla riga dove si ripropone la richiesta base (*riga 100*).

GOSUBroutines. Righe 500 ... 610; il contenuto di queste riche comprendono le istruzioni di Edit o stampa della situazione aggiornata della scacchiera, quella piccola in basso. Anche le *righe 1000 ... 1240* contengono le istruzioni di routine; qui avviene l'aggiornamento delle tabelline B e T e viene eseguito anche un ulteriore controllo della validità della mossa eseguita. Per la tabellina T il valore 5 corrisponde all'esistenza di una pallina nera mentre il valore zero si intende casellina vuota; invece per la tabella B il valore di –7 è uguale alla definizione di una casella irraggiungibile e parimenti il valore di –3 definirà una casella raggiungibile. La routine sita fra le *righe 1500 ... 1630* esegue un controllo di fine gioco. Queste istruzioni del programma esaminano tutti i contenuti dei vari elementi delle due tabelle B e T ed il controllo stabilisce se esiste ancora la possibilità di ulteriori mosse oppure al contrario l'impossibilità della continuazione del gioco. Solo in quest'ultimo caso viene Editato (stampato) la quantità delle palline rimaste e solo nell'eventualità che una sola fosse la superstite viene aggiunto un commento di meritate congratulazioni.

Conclusioni

L'occupazione in memoria di questo gioco prima del normale RUN è uguale a 2464 bytes; mentre durante l'esecuzione l'utilizzo totale corrisponde a 3436 bytes, ciò è dovuto principalmente dal riempimento dei dati nelle tabelle. Questo tipo di occupazione in memoria permette anche per chi è possessore di personal con limitata memoria di poter utilizzare questo programmino senza essere obbligati ad eseguire modifiche o tagli.

Il Listing o listato che viene proposto in queste pagine è direttamente tratto dalla stampa di un tabulato uscito dalla periferica-stampante del 2001 e quindi esente da errori.

10 PRINT "BUON DIVERTIMENTO"
20 END

HOMIC
il più grande centro italiano di microcomputer

propone NASCOM 1, lire 390.500

Nascom 1 è il potente microsistema modulare particolarmente indicato per controlli e applicazioni speciali. Nella versione Kit è molto indicato per sistemi personal e didattici.

NASCOM 1
Il sistema base venduto in KIT o montato comprende:
- Sistema operativo su EPROM 1K (NASBUG)
- 1K RAM + 1K Video RAM
- Interfacce TV, Cassette, RS 232
- ENTRATE/USCITE parallele (16 bit)
- Tastiera alfanumerica.

OPZIONI DISPONIBILI
- SCHEDA BUFFER transizione BUS Z80 - NASBUS
- SCHEDA ESPANSIONE DI MEMORIA: 8, 16, 32K RAM + 4 EPROM
- SCHEDA ESPANSIONE I/0: 1 CTC, 1 UART, 3 PIO
- ALIMENTATORE 3A

SOFTWARE DISPONIBILE
- ASSEMBLER EDITOR su cassetta (ZEAP)
- MONITOR 2K su EPROM (NASBUG T4)
- Tiny Basic 2K su EPROM
- Super Tiny Basic 3K su EPROM
- BASIC 8K su cassetta o su EPROM.

HOMIC
i "micro" in negozio.

Milano - uffici: Piazza De Angeli 1 - Tel. 4695467/4696040
centro vendite: Galleria De Angeli 1 - Tel. 437058

Distributori HOMIC

AZ ELETTRONICA ROBBIA
Via Varesina 205
Milano
Tel. 02/3086931

HOME DATA SYSTEM
Via Vercellotto 134/B
Cossato (Vc)
Tel. 015/93770

CO.R.EL.
Via Mercato Vecchio 28
Udine
Tel. 0432/291466

MICRODATASYSTEM
Via Vespasiano 56/B
Roma
Tel. 06/31600

DIGITRONIC
Via Provinciale 46
Tavernerio (Co)
Tel. 031/427076

K-BYTES
Via XX Settembre 20
Genova
Tel. 010/5926636

GVH - Gianni Vecchietti
Via Cipriani 18
Bologna
Tel. 051/279482

E. E. C.
Via La Farina 40
Messina
Tel. 090/2924164

PER L'AUTOMAZIONE INDUSTRIALE E IL CONTROLLO DI PROCESSO:

DAI THE MICROCOMPUTER

VERSATILE

- Sistema modulare su eurocard (100 × 160 mm)
- Moduli a microcomputer con elevatissima densità di funzioni
- Set completo di interfacce VERAMENTE industriali
- Modifica e ampliamento immediato del sistema senza riprogettazione

AFFIDABILE

- Schede testate in tutte le loro funzioni
- Garanzia totale di 1 anno
- Manutenzione immediata con la sostituzione delle schede
- Disponibilità da stock con assistenza dalla rete di distribuzione

LO STANDARD SU MISURA

- Realizzazione di qualsiasi sistema senza neccessità di approfondire la tecnologia del microprocessore
- Eliminata la possibilità di ridondanza grazie all'archittettura modulare
- Sistema di sviluppo potente ed economico con Monitor, Editor, Assembler, Go, Step, Programmatore di Eprom e, facoltativi, Basic o Fortran

EMESA S.P.A.

Via L. Da Viadana, 9 - 20122 Milano

Telefono (02) 8690616 - 860307

Telex 335066

Tester per circuiti integrati TTL realizzato con il Nanocomputer NBZ80-S

Parte I di **A. Cavalcoli**

In questo articolo, e nel successivo, descriveremo una interessante applicazione pratica del Nanocomputer: progettare un tester per circuiti integrati TTL a 14 o 16 pin.
Al fine di fornire tutte le informazioni necessarie al lettore, si premetterà una breve descrizione della versione Super del Nanocomputer NBZ80, cioè il sistema didattico NBZ80-S, e si discuteranno alcune considerazioni teoriche in merito al problema da risolvere.

Il Nanocomputer NBZ80-S

In Figura 1 è data la configurazione completa del sistema didattico NBZ80-S proposto dalla SGS-ATES.
Questa versione è semplicemente un'espansione del sistema Nanocomputer NBZ80, nella logica di compatibilità e di crescita modulare proposta dalla SGS-ATES.
Rispetto al sistema base NBZ80, la configurazione di Figura 1 presenta le seguenti unità addizionali:
- NEZ80, scheda per esperimenti su doppio eurocard, direttamente collegata alla scheda NBZ80 per l'attuazione di esperimenti di interfacciamento con il microprocessore Z80;
- NPZ80, contenitore ed alimentatore per il sistema Nanocomputer. Il contenitore è stato realizzato per ospitare due schede doppio eurocard (NBZ80 ed NEZ80), mentre l'alimentatore fornisce le alimentazioni $\pm 5V$, $\pm 12V$.

Sarà questa la versione del Nanocomputer utilizzata nell'esperimento in oggetto.

La scheda per esperimenti del Nanocomputer NBZ80-S

Diamo ora alcune indicazioni sulla scheda NEZ80, scheda per esperimenti del Nanocomputer (v. Figura 2).
Gli elementi costituenti sono:
- i connettori J1, J2, J3;
- un connettore PIO da 40 pin;
- una basetta di montaggio (breadboard) SK-10;
- otto indicatori luminosi, LM0, LM1, LM2, LM3, LM4, LM5, LM6, LM7;
- otto interruttori logici, SW0, SW1, SW2, SW3, SW4, SW5, SW6, SW7;
- due interruttori con molla di ritorno (pulser), P0 e P1;
- una spia di alimentazione (contrassegnata con +5V).

I connettori J1 e J2 mettono a disposizione i segnali presenti sul bus del Nanocomputer (Gamma BUS) ai circuiti montati dall'utente, tramite lo zoccolo SK-10 ed i tre zoccoli a 40 pin A, B e C.
Il connettore J3 è invece adibito all'alimentazione della scheda per esperimenti.
Il connettore PIO collega la scheda per esperimenti con il secondo PIO del Nanocomputer tramite un cavo flessibile a 40 poli, affinché i pin del PIO stesso possano essere riportati ai fori degli zoccoli A, B e C.
Si tenga presente che sulla scheda CPU del Nanocomputer vi sono due chip PIO, di cui il primo è adibito alle comunicazioni con l'unità tastiera-display, mentre il secondo è libero a disposizione dell'utente.
La basetta di montaggio SK-10 (breadboard) (v. Figura 3) è concepita in modo tale da permettere la realizzazione, senza necessità di saldature, di circuiti ideati dall'utente. Ai due lati di una scanalatura centrale, lo zoccolo presenta 64 gruppi di 5 terminali a non saldare.

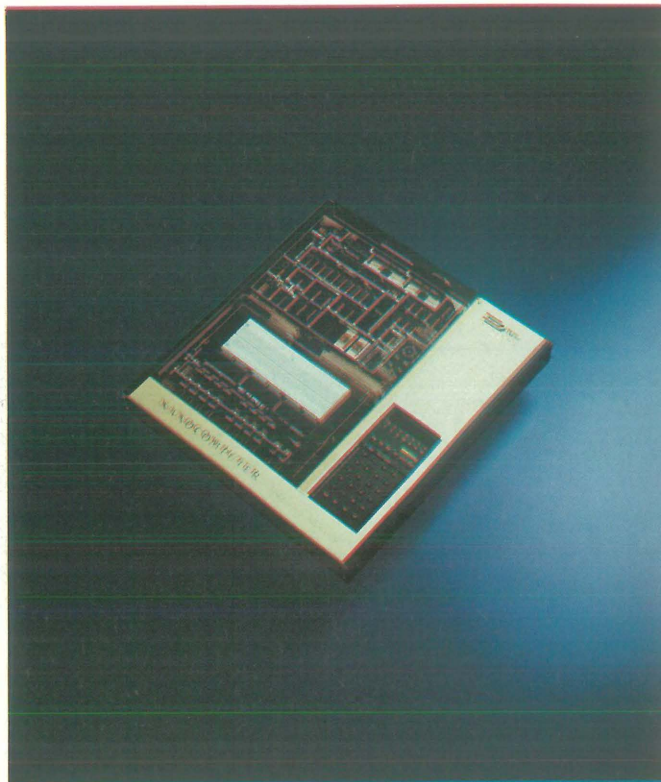



Figura 1 - Nanocomputer NBZ80-S della SGS-ATES.

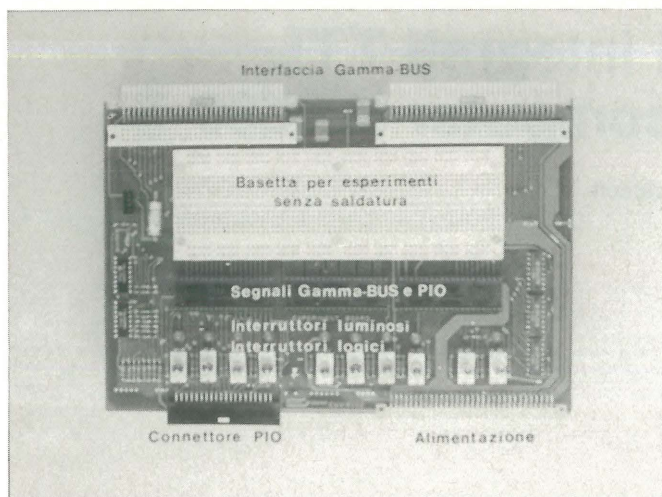



Figura 2 - Scheda di montaggio NEZ80 vista dall'alto.

I terminali di ogni gruppo sono collegati elettricamente.
Verso l'esterno vi sono 4 insiemi con 5 gruppi di 10 terminali pure collegati elettricamente.
I gruppi centrali servono per alloggiare i circuiti integrati, mentre i gruppi esterni sono adibiti all'alimentazione dei circuiti stessi.

Figura 3 - Basetta SK-10 vista dall'alto.

Infine, i connettori a 40 pin A, B e C sono di fatto degli zoccoli standard per circuiti integrati a 40 pin, collegati a J1, J2 ed al connettore PIO tramite le piste del circuito stampato della scheda per esperimenti.
Su questi zoccoli si ha un totale di 86 segmenti diversi, ciascuno dei quali può essere utilizzato nei circuiti montati sulla basetta SK-10.
Il dettaglio dei segnali è dato in Tabella I.
Sulla scheda per esperimenti sono poi presenti otto indicatori luminosi (LED), otto indicatori logici e due pulser, utilizzabili come dispositivi di ingresso.
Ogni interruttore determina uno stato 0 logico oppure 1 logico al proprio terminale di uscita (linee SWn, con n = 0 ... 7 sullo zoccolo A).
Analogamente ogni pulser determina la comparsa di un impulso ai due terminali di uscita Pn e $\overline{Pn}$ (n = 0,1).

Il circuito di I/O parallelo PIO

In appendice a questo articolo sulle applicazioni del Nanocomputer NBZ80, al fine di mettere a disposizione del lettore le informazioni necessarie per un proficuo utilizzo dell'applicazione descritta, diamo una descrizione sommaria del circuito di I/O parallelo PIO, prodotto dalla SGS-ATES.
Il PIO, chip a 40 pin, presenta le seguenti caratteristiche:
- *due porte bidirezionali ad otto bit;*
- *operazioni di ingresso/uscita pilotate da interrupt;*
- *quattro modi di funzionamento selezionabili:*
  - *a) modo 0: uscita di un byte*
  - *b) modo 1: ingresso di un byte*
  - *c) modo 2: trasferimento bidirezionale (solo porta A)*
  - *d) modo 3: controllo di bit singoli*

Per più dettagliate informazioni tecniche si faccia riferimento al Manuale Tecnico sul PIO Z80, pubblicato dalla SGS-ATES.
Nella Figura A1 è data la configurazione dei pin del chip PIO.
La Figura A2 presenta invece lo schema a blocchi funzionale del PIO.
Il bus dei dati viene utilizzato per il trasferimento di tutti i dati e comandi tra la CPU Z80 ed il PIO (CPU DATA BUS).
Le linee di controllo hanno le seguenti funzioni:

- CE	*abilitazione del PIO (Chip Enable)*
- C/D	*definizione del tipo di byte da trasferire (Controllo/Dato)*
- B/A	*selezione della porta (A o B)*
- $\overline{M1}$, $\overline{IORQ}$, $\overline{RD}$	*controllo del flusso dei dati tra il PIO e la CPU*
- $\overline{INT}$, $\overline{IEI}$, $\overline{IEO}$	*controllo interruzioni.*

Sono presenti infine le linee relative alle due porte (A e B) di ingresso/uscita:
- *linee dati e controllo per la porta A, cioè da A0 ad*

Figura A1 - Configurazione dei piedini del dispositivo PIO. (Per concessione della SGS-ATES).

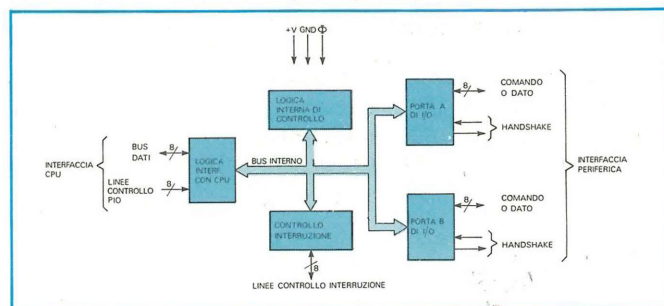



Figura A2 - Schema a blocchi funzionale del PIO. (Per concessione della SGS-ATES).

Tabella I - Segnali disponibili sugli zoccoli A, B e C suddivisi per categoria.

CPU Z80	39 segnali, uno per ogni pin dello Z80, ad esclusione di GND e Vcc, più un impulso BWR ritardato
PIO	20 segnali del PIO disponibile
Indicatori luminosi	Otto segnali di ingresso per i driver dei LED
SWITCH	Otto segnali di uscita dagli interruttori con logica antirimbalzo
Pulser	Quattro segnali impulsivi con logica antirimbalzo
Controllo di I/O	Nove segnali già decodificati per la selezione di dispositivi periferici
Alimentazione	Quattro linee: +5, –5, +12, GND

Con riferimento ai LED, questi sono accesi se l'ingresso corrispondente è ad 1 logico, spenti se a 0 logico.

Presentazione dell'applicazione

Elemento centrale dell'applicazione sarà il secondo PIO presente sulla scheda del Nanocomputer; esso sarà programmato per il modo 3 di funzionamento (Modo di Controllo) per entrambe le sue porte di I/O, con conseguente configurazione software delle sue 16 linee di I/O come linee di ingresso o di uscita. Si effettuerà un collegamento dei pin del chip TTL, a 14 o 16 pin, sotto test alle due porte del PIO.
Con un opportuno programma si porranno in uscita dei dati di stimolo e si leggeranno i dati risultanti dal chip sotto test.
I risultati saranno poi sottoposti ad analisi per verificarne la correttezza o meno.
Sostanzialmente si verificherà che la programmabilità del PIO determina un'elevata efficienza che, unitamente ad una notevole semplicità dei circuiti di interfaccia necessari, permette la realizzazione pratica del tester per integrati TTL.

Teoria dell'applicazione

Il circuito integrato sotto test è considerato alla stregua di un sistema digitale con più ingressi ed uscite di tipo combinatorio, in cui cioè la combinazione degli stati logici in ingresso determina i valori logici ai pin di uscita (basti pensare alle porte AND, OR, etc.).
Approfittiamo per ricordare che in dispositivi a logica sequenziale, quali flip-flop e contatori, sono significative invece anche le transizioni tra i livelli logici, oltre agli stati logici nei pin di ingresso.
Il metodo di test proposto è il seguente:
a) generare la tabella della verità relativa ad un chip

A7, con le linee di handshake ARDY e $\overline{ASTB}$
- linee dati e controllo per la porta B, cioè da B0 a B7, con le linee di handshake BRDY e $\overline{BSTB}$.
Per comprendere il funzionamento del PIO, è utile far riferimento alla Figura A3, che fornisce lo schema a blocchi di una porta di I/O (A o B).
Il Mode Control Register (Registro di Controllo Modo) contiene un codice a due bit che specifica il modo attuale di funzionamento del PIO (si ricorda che nell'applicazione in studio il PIO dovrà operare nel modo 3). Questo registro è caricato, da software, con un byte di comando dalla CPU verso il PIO.
Nel modo 3, e solo in questo, vengono usati i registri: Input/Output Select Register (Registro Selezione Ingresso/Uscita), Mask Control Register (Registro di Controllo Maschera), Mask Register (Registro Maschera).

Figura A3 - Schema a blocchi funzionale di una porta di I/O del PIO. (Per concessione della SGS-ATES).

Il modo 3 è caratterizzato dal fatto che ognuno degli 8 bit di ciascuna porta del PIO può essere specificato come bit di input o di output, e questa definizione viene realizzata dall'Input/Output Select Register (8 bit), il cui contenuto indica se una linea è di input (bit corrispondente ad 1 logico) o di uscita (0 logico).
Nel modo 3 è possibile poi il riconoscimento di una data configurazione di bit, nel qual caso il PIO genera un'interruzione verso lo Z80.
Per definire la configurazione voluta si utilizzano a tal fine il Mask Register (8 bit) ed il Mask Control Register (2 bit).
Per una più completa descrizione, si veda il Nanobook III, di prossima pubblicazione da parte della Jackson Italiana.
A questo punto resta da vedere come è possibile, in pratica, scegliere per il PIO un certo modo di funzionamento.
Questo è realizzato inviando, come parola di controllo verso la porta desiderata del PIO, un byte così configurato:

D7	D6	D5	D4	D3	D2	D1	D0
M1	M0	X	X	1	1	1	1

dove è:

M1	M0	SIGNIFICATO
0	0	Modo 0
0	1	Modo 1
1	0	Modo 2
1	1	Modo 3

Selezionando il modo 3 (M0 = 1, M1 = 1), il PIO interpreterà il successivo byte di controllo come un'informazione diretta al Registro Input/Output Select, che definisce quali linee sono d'ingresso e quali di uscita.

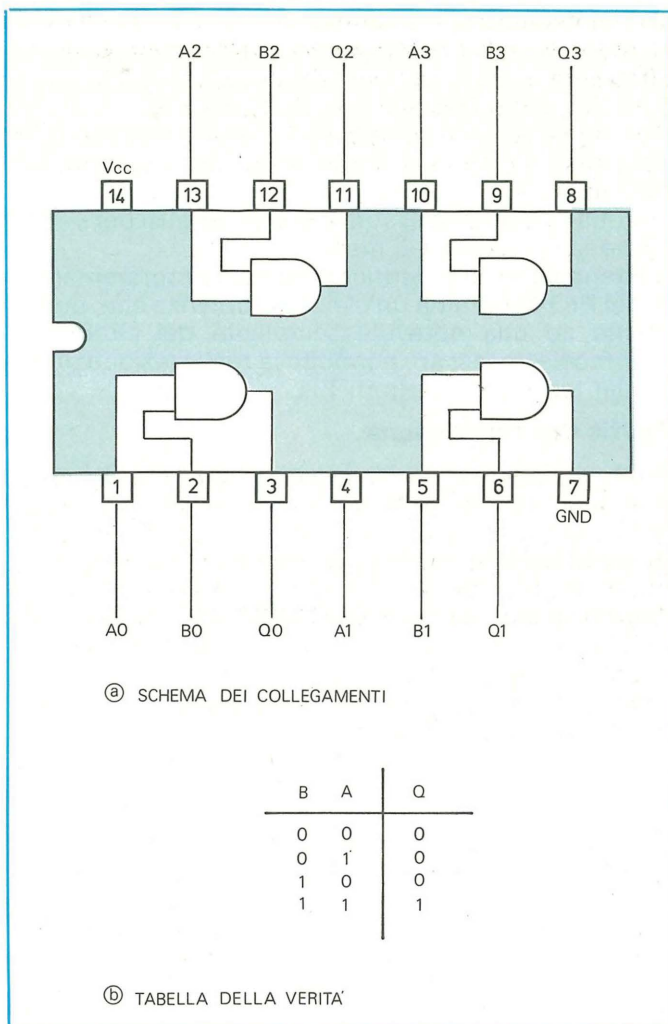



Figura 4 - Un Quad AND Gate a due ingressi 74LS08.

campione, sul cui corretto funzionamento si è sicuri;

b) generare la tabella della verità del chip sotto test;

c) confrontare le due tabelle della verità.

Come spiegazione ulteriore, si consideri il chip 74LS08, AND quadruplo a due ingressi di Figura 4.

Il chip si può considerare costituito da quattro porte AND indipendenti, ciascuna con due ingressi ed una sola uscita, cosicché ad ogni porta si può associare una tabella della verità.

Alternativamente, si può vedere il 74LS08 come un circuito combinatorio più complesso, con 8 ingressi e 4 uscite (v. Figura 5). Da cui una tabella della verità ad 8 ingressi e 4 uscite, una parte della quale è data in Figura 6.

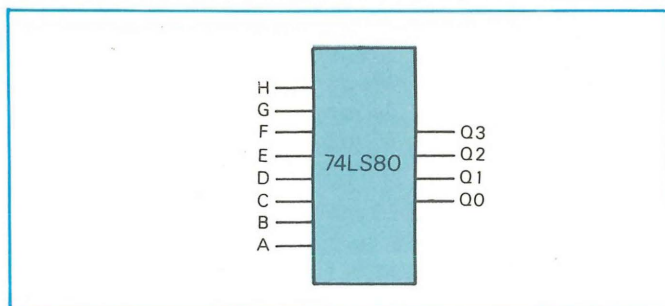



Figura 5 - Rappresentazione alternativa del 74LS08.

Inputs								Outputs			
H	G	F	E	D	C	B	A	Q0	Q1	Q2	Q3
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0		0	0
0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	1	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1	0	0	0	0
0	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	1	0	1	1	1	0	0	0
0	0	0	0	1	1	0	0	0	1	0	0
0	0	0	0	1	1	0	1	0	1	0	0
0	0	0	0	1	1	1	0	0	1	0	0
0	0	0	0	1	1	1	1	1	1	0	0
.									.		
.									.		
.									.		
1	1	1	1	1	1	1	1	1	1	1	1

Figura 6 - Tabella Generale della Verità che tratta il circuito integrato 74LS08 come un sistema combinatorio dotato di otto ingressi e quattro uscite.

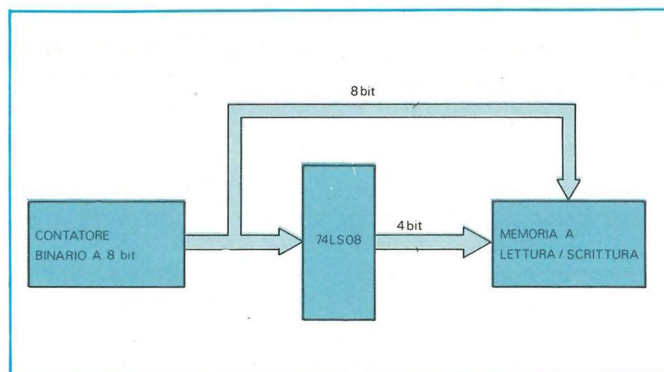



Figura 7 - Componenti funzionali di un tester di circuiti integrati.

In quest'ultimo modo si presenta una descrizione delle caratteristiche del chip molto utile ai fini della sua verifica funzionale con il tester.

La Figura 7 fornisce una schematizzazione degli elementi necessari per un tester convenzionale: con un contatore ad 8 bit si applicano via via 256 differenti combinazioni di ingresso al chip 74LS08.

Le uscite (4 bit) sono contemporaneamente memorizzate in 256 locazioni di memoria.

Operazione finale deve essere quella di confrontare queste risposte con la "tabella campione".

Scopo della presente applicazione sarà quello di sostituire l'hardware indicato in Figura 7 con il chip PIO del Nanocomputer e con il software necessario sia a porre in uscita verso il chip sotto test dei byte di valore crescente che a memorizzare i valori di risposta.

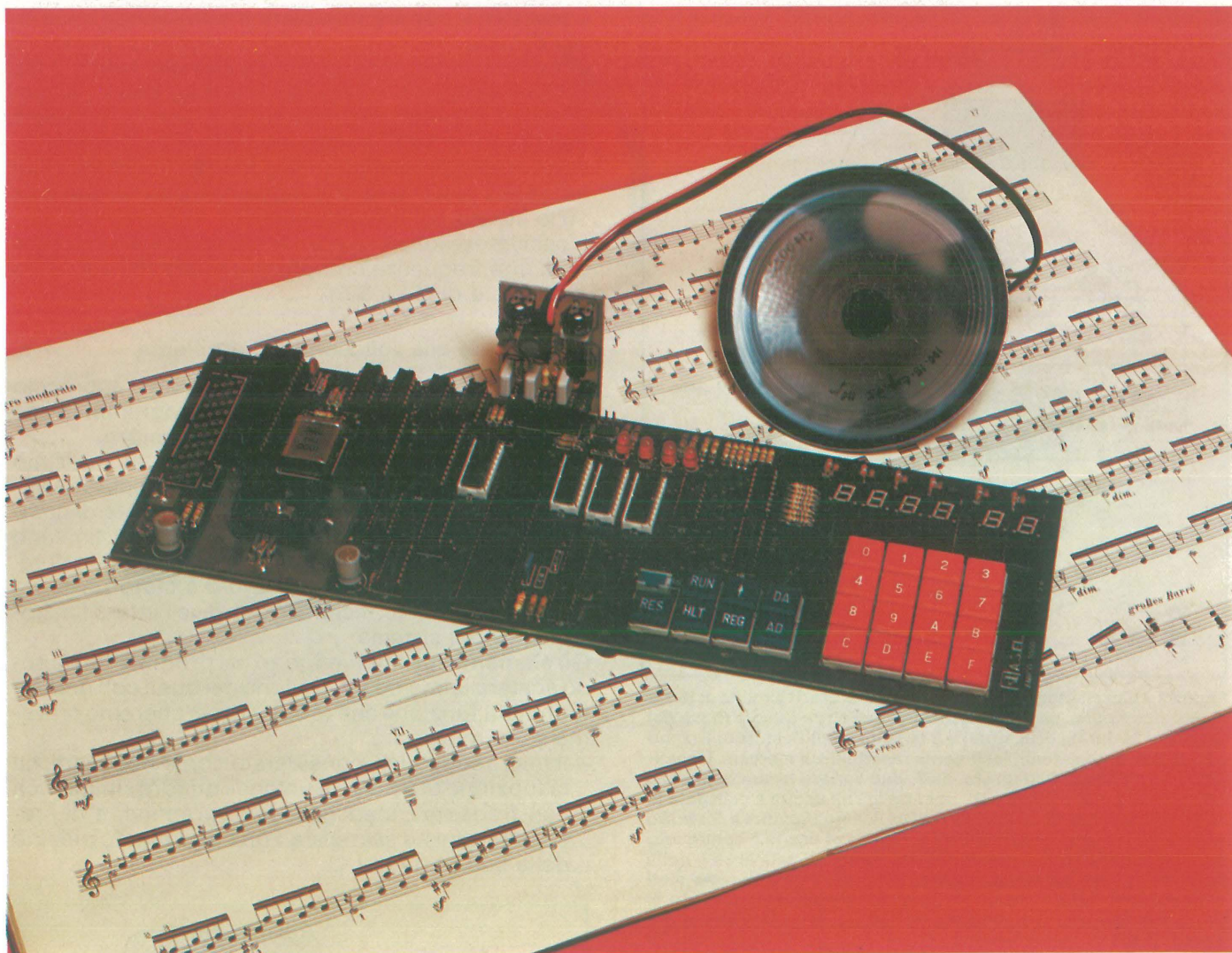
Per una corretta configurazione del tester sarà necessario conoscere, per il circuito integrato sotto esame, i seguenti dati:

a) la posizione dei pin di alimentazione;

b) la posizione di tutti i pin di ingresso ed uscita;

c) i pin per cui non sono previsti dei collegamenti (che devono essere posti a +5 o a massa).

Si noti che il chip non deve essere né un dispositivo analogico-digitale la cui operazione sia legata a circuiti RC di temporizzazione, né un dispositivo i cui ingressi siano attivi su fronti.

■

Il microelaboratore nella composizione musicale

di **A. Tramontini**

Un microelaboratore può essere considerato come un mezzo in grado di suonare, con tempi e ritmi esatti, la musica scritta su uno spartito musicale. In fondo si tratta di gestire numeri: frequenze, durata dei suoni, pause. Semmai il problema vero non sta tanto nelle possibilità di un sistema di elaborazione (che sono direttamente proporzionali al suo costo) quanto piuttosto nell'abbordabilità economica dello stesso. La soluzione qui presentata fa riferimento ad un hardware alla portata di tutti, immediatamente disponibile sul mercato e affidabile, e ad un software che, compatibilmente con la memoria richiesta, dà una serie di prestazioni tali da essere considerate soddisfacenti.

Caratteristiche del sistema

Vediamo dunque di definire le prestazioni del sistema che ci proponiamo di realizzare dal punto di vista della "programmabilità" musicale.

Altezza dei suoni: 6 ottave, di cui 2 in chiave di basso e 4 in chiave di violino.
Numero massimo di note e pause: 500, disponendo di una memoria RAM da 1 Kbyte.
Durata delle note o pause: 4/4, 2/4, 1/4, 1/8, 1/16.
I componenti con i quali sono state ottenute queste prestazioni sono:
- il microelaboratore AMICO 2000
- un oscillatore ad onda quadra

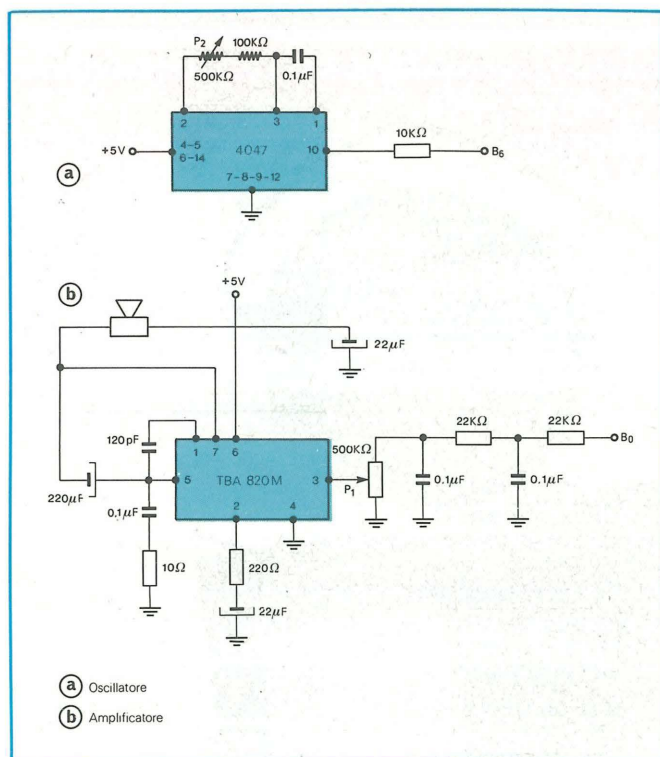



Figura 1 - L'oscillatore esterno e l'amplificatore. Il circuito esterno al microcomputer per la generazione dell'onda quadra (base dei tempi per la durata delle note) e il circuito amplificatore utilizzano pochissimi componenti, facilmente reperibili sul mercato. L'oscillatore, che utilizza l'integrato 4047, può variare la sua frequenza agendo sul potenziometro P2, mentre la sua uscita è collegata al bit 6 della porta di I/O dell'AMICO 2000. L'amplificatore è costruito intorno all'integrato TBA 820 M; il volume di uscita è controllato dal potenziometro P1, mentre l'uscita sarà collegata al bit 0 della porta di I/O del microelaboratore. Entrambi i circuiti sono stati montati dalla A.S.E.L. su un unico circuito stampato provvisto di un connettore per il collegamento diretto al microelaboratore AMICO 2000.

- un amplificatore audio.

L'oscillatore e l'amplificatore sono montati su uno stesso circuito stampato.

Il programma che descriveremo in seguito è stato sviluppato su questa macchina, ma è molto semplice adattarlo ad altri microelaboratori che usano il 6502 come CPU, oppure riscrivere il tutto anche per altri tipi di CPU.

L'oscillatore ad onda quadra e l'amplificatore, i cui schemi appaiono in Figura 1, sono realizzati con pochissimi elementi. L'insieme, per essere utilizzata con l'AMICO 2000, è stato montato su un piccolo circuito stampato completo di connettore per inserzione diretta sulla porta di I/O parallela da 8 bit presente sullo stesso microcomputer.

L'oscillatore funziona con una tensione di 5 Vcc e genera una frequenza compresa fra 20 e 70 msec. Con duty cycle di circa 50%.

L'approccio alla soluzione del problema

Vediamo ora a grandi linee quale è stato l'approccio al problema della generazione e programmazione delle note musicali con il microelaboratore.

Innanzitutto si è cercato di utilizzare al massimo l'hardware esistente, ovvero il microcomputer AMICO 2000, riducendo al minimo la necessità di componenti esterni, che, come abbiamo visto, si limitano ad un semplicissimo oscillatore ed un amplificatore.

Pertanto il grosso carico di lavoro è stato trasferito sul software, cui praticamente si deve l'intero funzionamento del sistema.

Gli elementi che intervengono nel progetto finale sono sostanzialmente tre, ognuno dei quali con una determinata funzione che di seguito cercheremo brevemente di analizzare:

- il microcomputer, considerato come hardware, ha la funzione di generare un'onda quadra simmetrica con freqeunza stabilita dal programma, e di presentare questo segnale ad un determinato piedino della sua porta di I/O.

```
      MAIN

0000  A9 98              LDA  #$98    PREDISPONE I/O
   2  8D 03 FD           STA  PORTD
   5  20 4C 02           JSR  BIT     BASE INDICE TABELLA
   8  4C 0E 00           JMP  B
   B  20 32 02   A       JSR  IIT     INCREMENTO INDICE TABELLA
   E  20 55 02   B       JSR  AND     ACQUISIZIONE NOTA DURATA
  11  A5 ED              LDA  NOTA
   3  D0 06              BNE  C
   5  20 6F 02           JSR  PAUSA   PAUSA
   8  4C 0B 00           JMP  B
   B  20 06 02   C       JSR  OUTN    OUT NOTA
   E  A2 FF              LDX  #$FF    RITARDO 1MSEC
  20  CA         D       DEX
   1  D0 FD              BNE  D
   3  4C 0B 00           JMP  A
```

Locazioni utilizzate

00ED = NOTA	Deposito valore Note

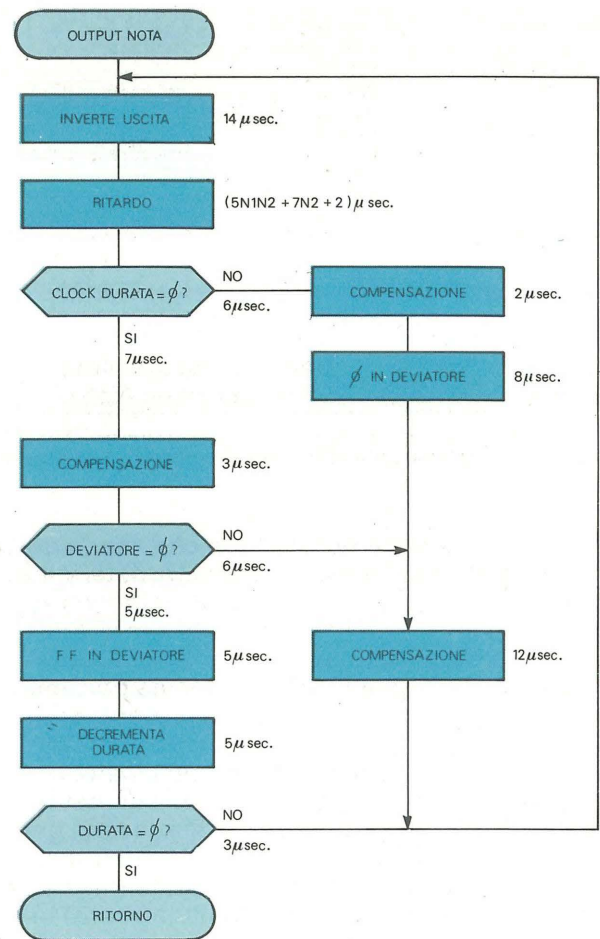
Figura 2 - Listato e flow del programma principale di generazione delle note.

```
        OUTPUT NOTA PER RELATIVA DURATA

0200   B5 EA      A1    LDA DEP,X    12 MICROSEC DI COMPENSAZIONE
   2   EA EA            NOP
   4   EA EA            NOP
   6   E6 EF      A     INC OUTN     INVERT OUT
   8   A5 EF            LDA OUTN
   A   29 01            AND #$01
   C   8D 03 FD         STA PD       OUT STORE IN PORT D
   F   A4 EE            LDY MNOTA    (N2)
  11   A6 ED      A3    LDX NOTA     (N1)
   3   CA         A2    DEX          (5N1N2+7N2+2)DELAY
   4   DO FD            BNE A2
   6   88               DEY
   7   DO F8            BNE A3
   9   2C 02 FD         BIT PB
   C   50 07            BVC B        IF BIT 6 = 0 IN B
   E   EA               NOP          ELSE 0 IN DEVIATORE
   F   A9 00            LDA #$00
  21   85 EB            STA DEV
   3   FO DB            BEQ A1
   5   A5 EB      B     LDA DEV      (COMPENSAZIONE)
   7   A5 EB            LDA DEV
   9   DO D5            BNE A1       IF DEVIATORE <> 0 IN A1
   B   C6 EB            DEC DEV      ELSE $ FF IN DEVIATORE
   D   C6 EC            DEC DUR      DURATA=DURATA-1
   F   DO D5            BNE A        IF>0 IN A
  31   60               RTS
```

Locazioni utilizzate

00EA = DEP	Deposito Generico
00EB = DEV	Deviatore
00EC = DUR	Deposito Valore Durata
00ED = NOTA	Deposito Valore Note
00EE = MNOTA	Moltiplicazione Note
OOEF = OUTN	Deposito Uscita

Figura 3 - Listato e flow della subroutine di pilotaggio dell'amplificatore.

- l'oscillatore esterno ad onda quadra, cha ha la funzione di base dei tempi e determina la durata delle note, costituisce il secondo elemento del sistema ed è in comunicazione con il computer ancora tramite la porta di I/O: modificando la frequenza di questo segnale esterno si ha in corrispondenza una più o meno lunga durata dell'1/16 del tempo base.
  La definizione dell'unità di tempo (1/16 di battuta) di questo oscillatore corrisponde in un certo senso al tempo che imposteremmo sul metronomo per determinare la velocità con la quale vogliamo che un certo brano venga eseguito (largo, adagio, allegro, etc.);
- il software, ovvero il programma di gestione del tutto, è il terzo e più importante elemento del sistema. In particolare, tramite software ci si occupa della gestione della frequenza della nota, che è ottenuta

```
        INCREMENTA INDICE TABELLA,BASE INDICE TABELLA

0232   D8               CLD          CLEAR DECIMAL MODE
   3   18               CLC          CLEAR CARRY
   4   A5 DO            LDA INDL
   6   69 02            ADC #$02
   8   85 DO            STA INDL
   A   A5 D1            LDA INDH
   C   69 00            ADC #$00
   E   85 D1            STA INDH
  40   C5 D3            CMP FINEH
   2   90 10            BCC OUT
   4   DO 06            BNE A
   6   A5 DO            LDA INDL
   8   C5 D2            CMP FINEL
   A   90 08            BCC OUT
   C   A9 00      A     LDA #$00     BASE INDICE TABELLA
   E   85 DO            STA INDL
  50   A9 04            LDA #$04
   2   85 D1            STA INDH
   4   60         OUT   RTS
```

Locazioni utilizzate

00D0 = INDL 00D1 = INDH	INDICE TABELLA
00D2 = FINEL 00D3 = FINEH	FINE TABELLA

Figura 4 - Listato e flow della subroutine di scansione del testo musicale.

```
        ACQUISISCE NOTA E DURATA

0255   A0 01          LDX #$01          1 IN Y
   7   B1 D0          LDA (INDH,L)Y     ACQUISISCE N1
   9   85 ED          STA NOTA
   B   88             DEY
   C   B1 D0          LDA (INDH,L)Y     ACQUISISCE N2
   E   29 0F          AND #$ 0F
  60   85 EE          STA MNOTA
   2   B1 D0          LDA (INDH,L)Y     ACQUISISCE DURATA
   4   29 F0          AND #$F0
   6   4A             LSR
   7   4A             LSR
   8   D0 02          BNE A
   A   09 40          ORA #$40
   C   85 EC     A    STA DUR
   E   60             RST
```

Locazioni utilizzate

00EC = DUR	Deposito Durata
00ED = NOTA	Deposito Valore Note
00EE = MNOTA	Moltiplicazione Note

Figura 5 - Listato e flow della subroutine di acquisizione della nota e della durata.

mediante un loop. È questo loop che, mediante la variabilità di alcuni dei suoi parametri, determina la durata del periodo dell'onda quadra simmetrica in funzione della frequenza della nota che si vuol generare.
In pratica l'onda quadra viene ottenuta portando a 0 e ad 1 e viceversa il livello logico presente in uscita al bit 6 della porta di I/O del microelaboratore. Il segnale così ottenuto viene poi amplificato. Il programma di gestione inoltre si occupa di mantenere, mediante un'apposita subroutine, la nota generata per tanti 1/16esimi di battuta quanti sono quelli assegnati dal programma utente (fino ad un massimo di 16/16 = 4/4 = una battuta intera) in relazione al tempo unitario scandito esternamente (e predeterminato dall'utente) dall'oscillatore ad onda quadra (base dei tempi).
Come vedremo infatti più in dettaglio, ogni nota viene definita da 2 byte: uno serve a generare il ritardo del loop, ovvero l'altezza (frequenza) della nota vera e propria, l'altro byte serve a stabilire il numero di 1/16esimi di mantenimento (durata della stessa nota o pausa).

Struttura del programma

Passiamo ora ad analizzare nei dettagli il programma di gestione del sistema, per dare anche le specifiche necessarie per "scrivere" delle note e/o pause nella memoria del nostro microcomputer.

Dei due Kbyte di RAM dell'AMICO 2000 usiamo le locazioni libere dalla 0400 alla 07FF (ogni nota o pausa occupa due locazioni di memoria successive) per memorizzare il testo musicale.
Il programma non permette di programmare durate superiori ai 4/4, né inferiori ad 1/16, mentre per la separazione uditiva di due o più note uguali viene introddotto in ritardo di 5 ms.
Il programma consta di una parte principale MAIN (v. Figura 2) che risiede in memoria da 0000 a 0025. Esso richiama 4 subroutines (fra parentesi sono riportate le locazioni di memoria occupate da ciascuna subroutine):

```
        PAUSA

026F   A9 00          LDA #$00      0 IN BIT 0 PORT CL
  71   8D 03 FD       STA PD
   4   2C 02 FD  A    BIT PC        TEST BIT 6 PORT CH
   7   50 06          BVC B         IF =0 IN B
   9   A9 00          LDA #$00      ELSE 0 IN DEVIATORE
   B   85 EB          STA DEV
   D   F0 F5          BEQ A
   F   A5 EB     B    LDA DEV       IF DEV <> 0 IN A
  81   D0 F1          BNE A
   3   C6 EB          DEC DEV       ELSE 1 IN DEV
   5   C6 EC          DEC DUR       DURATA=DURATA-1
   7   D0 EB          BNE A         IF > 0 IN A
   9   60             RST
```

Locazioni utilizzate

00EB = DEV	Deviatore
00EC = DUR	Deposito Durata.

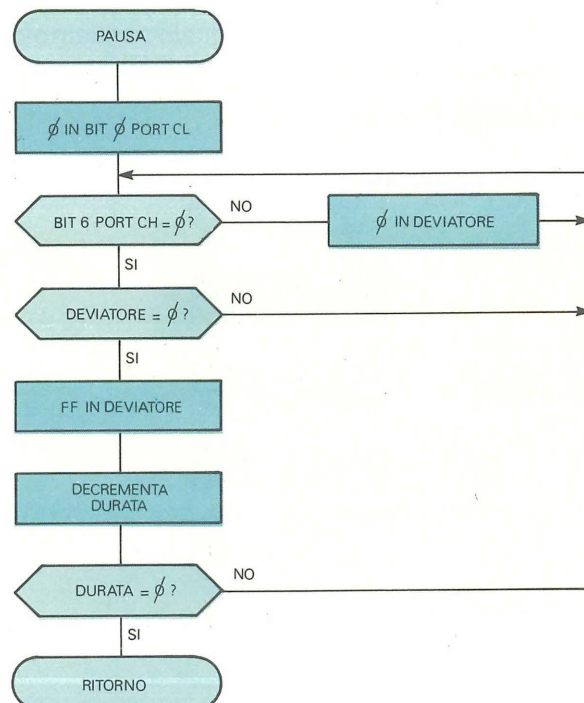



Figura 6 - Listato e flow della subroutine di pausa.

Figura 7 - Significato dei tasti dell'AMICO 2000 con il programma di edit musicale.

- output nota (0200 ÷ 0231) (v. Figura 3);
- incremento indice tabella (0232 ÷ 0254) (v. Figura 4);
- acquisizione nota e durata (0255 ÷ 026E) (v. Figura 5);
- pausa (026F ÷ 0289) (v. Figura 6).

Prima di scendere nei particolari del programma, vediamo quali sono le procedure per scrivere nella memoria le note e/o le pause e la loro durata: le spiegazioni che seguono servono a comprendere a fondo il funzionamento del programma fondamentale che genera le note e pause.
Innanzitutto il valore da attribuire ad ogni nota e la sua altezza (posizione sul rigo musicale) viene definito da due byte scritti in codice esadecimale, che vengono riportati nella Tabella I, a cui bisogna fare riferimento per comprendere gli esempi che seguono.

La programmazione della nota o pausa di data durata viene effettuata sommando il valore della nota o pausa al valore della durata.
Esempio:

Do 2ª ottava viol. da 1/4 sarà:

02B9 +
4000 =
____
42B9

in cui 4 = durata 1/4, e 2B9 = Do 2ª ottava violino.

È possibile sommare le durate per ottenere ad esempio la nota col punto o la legatura di due note uguali di durate diverse.
Esempi:

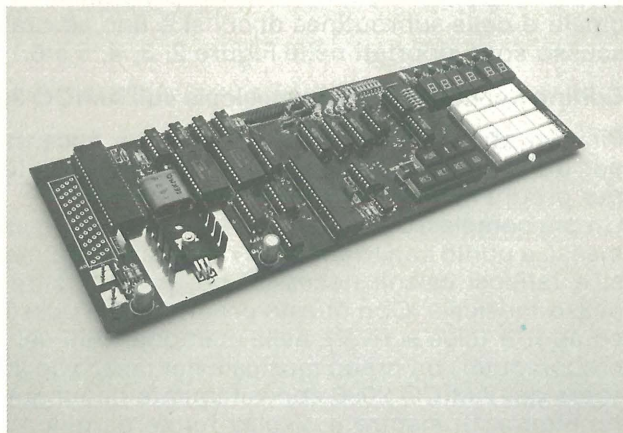
Mi 2ª ottava viol. da 2/4 col punto sarà:

0292 +
8000 +
4000 =
____
C292

Sol 3ª ottava viol. da 1/4 legato con Sol 3ª ottava viol. da 1/16 sarà:

0175 +
4000 +
1000 =
____
5175

Il microelaboratore AMICO 2000

L'AMICO 2000 è un microelaboratore progettato e costituito interamente in Italia dalla società A.S.E.L. s.r.l. di Milano.
Il microcomputer utilizzato nel progetto presentato in questo articolo è la scheda base del sistema, che contiene la CPU, il microprocessore 6502, per cui non occorre nessun altro elemento del sistema AMICO 2000.
Le caratteristiche fondamentali della scheda base sono:
- *il microprocessore 6502*
- *2 Kbyte di RAM utente*
- *1 Kbyte di ROM sul quale risiedono il programma di monitor del microelaboratore e quello di gestione dell'interfaccia cassette.*

- *una porta di input/output parallela da 8 bit per comunicare con l'esterno*
- *il circuito di interfaccia per registratore a cassette*
- *la tastiera esadecimale per l'introduzione dei dati e delle istruzioni e la tastiera funzionale per comandi di ingresso dati, avanzamento program counter, etc.*
- *un display a 6 digit LED per visualizzazione di dati e indirizzi*
- *un regolatore di tensione su scheda*
- *un generatore di clock a quarzo da 1 MHz.*

La scheda è provvista di un connettore per l'espansione del sistema, ovvero per il collegamento di altre schede, come quelle di interfaccia video, di espansione RAM o ROM, etc.
L'AMICO 2000 è stato presentato sulla rivista "Sperimentare" con una serie di articoli estremamente comprensibili che fra breve saranno raccolti in un unico testo
Così come è stata descritta, la scheda base AMICO 2000 viene venduta, montata e collaudata al prezzo di Lit. 285.000 più I.V.A. dalla A.S.E.L., Via Cortina d'Ampezzo, 17, Milano.
Il programma di editing musicale, presentato nell'articolo, è disponibile su cassetta insieme al listing completo e commentato. Sulla stessa cassetta è registrato anche un brano musicale, che ha una funzione di collaudo: per la cronaca, si tratta di una Bourrèe in SIb maggiore di R. de Visée.
Gli interessati potranno farne richiesta sempre alla A.S.E.L., insieme al generatore di onda quadra/amplificatore da collegare alla porta di I/O dell'AMICO 2000. Il prezzo di vendita complessivo è di Lit. 40.000.

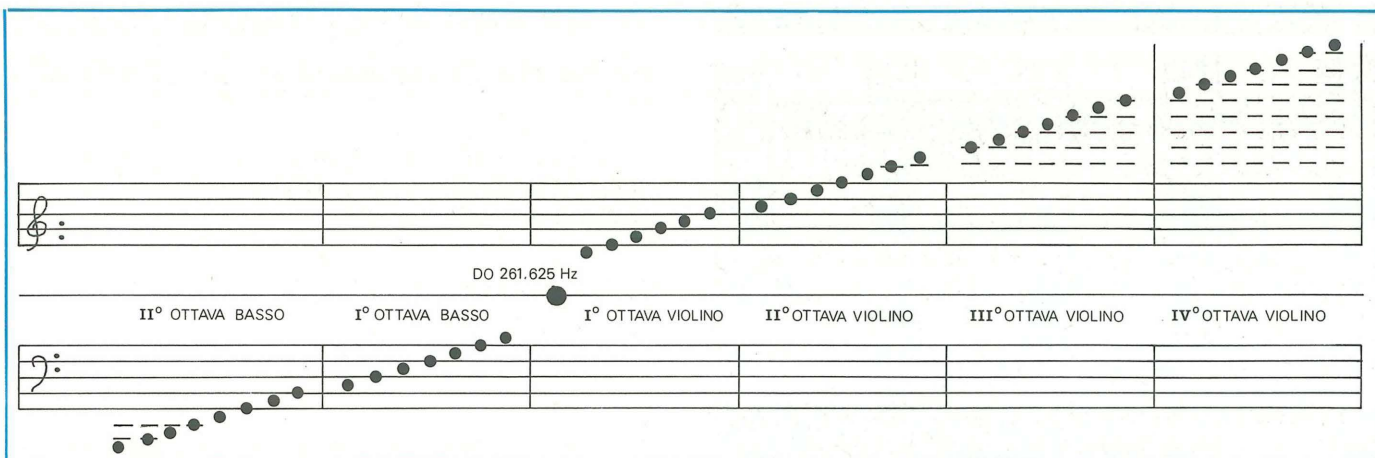



Figura 8 - Identificazione delle ottave da tastiera.

Una pausa di 1/1 sarà

0000 (valore pausa)
0000 (valore di 4/4)

0000

Memorizzazione del brano musicale

La prima nota o pausa programmata dovrà risiedere nelle locazioni 0400 e 0401, la seconda nelle locazioni 0402 e 0403, etc., in cui nella locazione più bassa è posta la parte alta delle note programmate, e nella successiva locazione la parte bassa delle note programmate.

Ad esempio, se la prima nota è Do 2ª ottava viol. da 1/4, il valore sarà 42B9 e risiederà in:

0400	42
0401	B9
	"
	"
	"
	"
	etc.

Inoltre nella locazione 00D2 dovrà essere posto il valore della parte bassa dell'indirizzo successivo all'ultimo utilizzato nella programmazione della tabella relativa al brano che s'intende far seguire.
Nella locazione 00D3 verrà introdotta la parte alta del suddetto indirizzo.

TABELLA VALORI DELLE NOTE

	2 OTTAVA BASSO	1 OTT BASSO	1 OTT VIOL.	2 OTT VIOL.	3 OTT VIOL.	4 OTT VIOL.
DO	0CFD	06FC	03FA	02B9	01B5	0155
REB/DO#	0CEE	06EE	03EC	02AE	01AA	0150
RE	0BF5	06E0	03DE	02A2	01A0	014B
MIB/RE#	0AFF	05FE	03D1	029B	0196	0146
MI	0AF0	05F0	03C5	0292	018D	0141
FA	09FC	05E2	03BA	0289	0185	013D
SOLB/FA#	09EE	05D5	03AF	0281	017D	0139
SOL	08FD	04FC	02F9	01F5	0175	0135
LAB/SOL#	08EE	04ED	02EB	01E6	016E	0132
LA	08E1	04E0	02DD	01D9	0167	012E
SIB/LA#	07F3	04D3	02D1	01CC	0161	012B
SI	07E5	04C7	02C5	01C0	015B	0128

VALORI DELLE DURATE

4/4 (16/16) = 0000
2/4 (8/16) = 8000
1/4 (4/16) = 4000
1/8 (2/16) = 2000
1/16 = 1000

VALORE DELLA PAUSA = 0000

Tabella I - Tabella dei valori delle note e delle durate.

Il numero massimo di note programmabili dipende dal numero di locazioni di memoria disponibili. Sull'AMICO 2000, partendo dalla locazione 0400, si arriva alla 07FF, ovvero 1024 locazioni. Poiché ogni nota occupa 2 locazioni, si potranno programmare fino a 512 note o pause di diversa durata.

Come si genera una nota

Considerando che il tempo di esecuzione della subroutine output nota è $(5N1 \cdot N2 + 7N2 + 44)$ μsec., che la frequenza del Do 2ª ottava violino è 261,625 Hz, e che il rapporto tra la frequenza di una nota e la precedente o la successiva è $\sqrt[12]{2} = 1{,}059$, si ricavano i valori N2 ed N1.
Questi valori, rappresentati con due caratteri esadecimali scritti nella sequenza N2 N1, rappresentano il valore del tempo di 1 semiperiodo della frequenza della nota.
(N2 viene anche definito come MNOTA: Moltiplicazione nota);
(N1 viene anche definito come NOTA).
Esempio:
Il Do di 261,625 Hz risulta uguale a 02 B9, essendo N2 = 02 ed N1 = B9.
Infatti:

$$\text{Freq.} = \frac{1}{2 \cdot (5N1N1 + 7N2 + 44)\,\mu\text{sec.}} =$$

$$\frac{1}{2 \cdot (5 \cdot 185 \cdot 2 + 7 \cdot 2 + 44)\,\mu\text{sec.}} = 262{,}055 \text{ Hz}$$

I flow charts con i relativi listati del programma principale e delle subroutines di cui si è fino ad ora discusso sono riportati nelle Figure 2, 3, 4, 5 e 6.

Editing per composizione musicale sull'AMICO 2000

Fin qui abbiamo esaminato il software necessario per generare note e pause con il microcomputer e per programmare l'esecuzione di un brano.
Un più complesso programma di editing però è stato messo a punto con lo scopo di risparmiare i numerosi e faticosi calcoli necessari per memorizzare un pezzo musicale. Con questo programma è possibile comporre (cioè scrivere nella memoria RAM del microcomputer) un brano musicale per mezzo della tastiera dell'AMICO 2000 appositamente riconfigurata (v. Figura 7), mentre il display funge da guida alle operazioni che vengono implementate.

love story

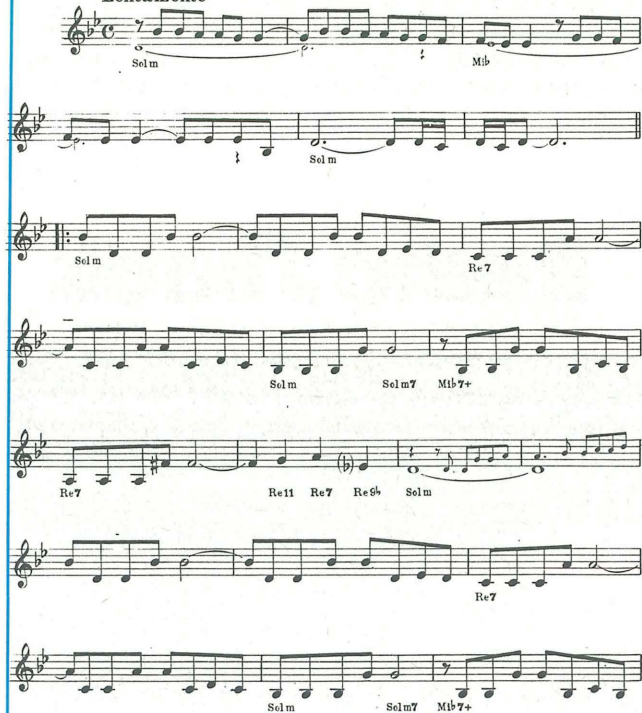
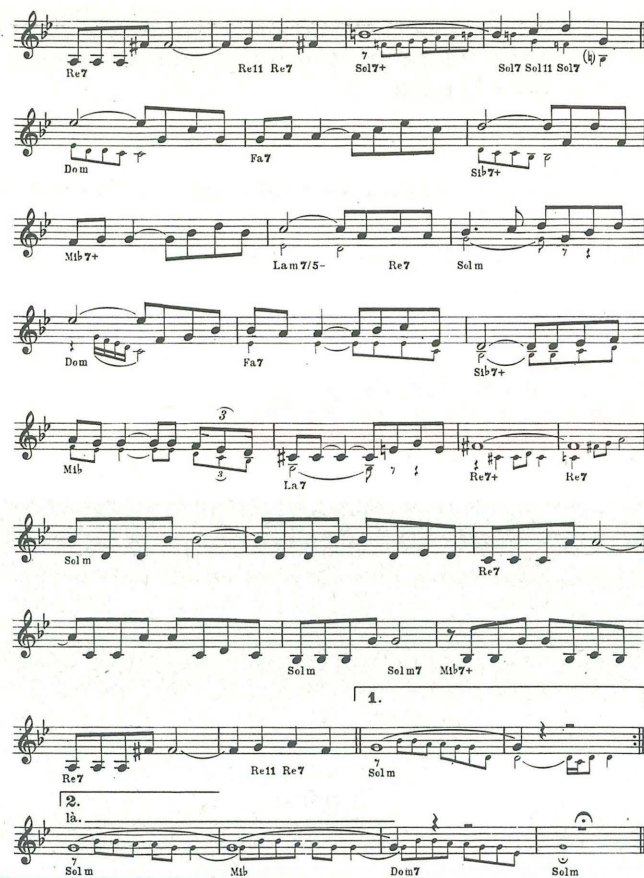
Lentamente

```
LOVE STORY

0400  20 00 21 CC 21 CC 21 D9 21 D9 21 F5 61 F5 21 CC
0410  21 CC 21 D9 21 D9 21 F5 21 F5 22 89 22 89 22 9B
0420  42 9B 20 00 21 F5 21 F5 22 89 22 89 22 9B 62 9B
0430  22 9B 22 9B 22 D1 E2 A2 12 A2 12 B9 12 A2 12 B9
0440  E2 A2 21 CC 22 A2 22 A2 21 CC A1 CC 22 A2 22 A2
0450  21 CC 21 CC 22 A2 22 9B 22 A2 22 B9 22 B9 22 B9
0460  21 D9 A1 D9 22 B9 22 B9 21 D9 21 D9 22 B9 22 A2
0470  22 B9 22 D1 22 D1 22 D1 21 F5 81 F5 20 00 22 D1
0480  22 D1 21 F5 21 F5 22 D1 22 B9 22 D1 22 DD 22 DD
0490  22 DD 22 81 C2 81 41 F5 41 D9 42 9B 42 A2 20 00
04A0  22 A2 22 A2 21 F5 21 F5 21 D9 61 D9 21 CC 21 CC
04B0  21 B5 21 B5 21 A0 21 CC 22 A2 22 A2 21 CC A1 CC
04C0  22 A2 22 A2 21 CC 21 CC 22 A2 22 9B 22 A2 22 B9
04D0  22 B9 22 B9 21 D9 A1 D9 22 B9 22 B9 21 D9 21 D9
04E0  22 B9 22 A2 22 B9 22 D1 22 D1 22 D1 21 F5 81 F5
04F0  20 00 22 D1 22 D1 21 F5 21 F5 22 D1 22 B9 22 D1
0500  22 DD 22 DD 22 DD 22 81 C2 81 41 F5 41 D9 42 81
0510  01 C0 41 C0 41 B5 41 A0 41 F5 A1 96 21 F5 21 B5
0520  21 F5 21 F5 21 D9 61 D9 21 B5 21 96 21 B5 A1 A0
0530  22 89 21 A0 22 89 22 89 21 F5 61 F5 21 CC 21 A0
0540  21 CC A1 B5 21 D9 21 B5 21 D9 61 CC 21 B5 21 A0
0550  21 F5 21 CC 21 A0 A1 96 22 89 21 F5 21 CC 21 CC
0560  21 D9 61 D9 21 CC 21 B5 22 9B A2 A2 22 A2 22 9B
0570  22 89 21 D9 21 F5 61 F5 21 F5 12 89 22 9B 12 A2
0580  22 AE 82 AE 22 92 21 F5 22 92 02 81 82 81 82 81
0590  21 CC 22 A2 22 A2 21 CC A1 CC 22 A2 22 A2 21 CC
05A0  21 CC 22 A2 22 9B 22 A2 22 B9 22 B9 22 B9 21 D9
05B0  A1 D9 22 B9 22 B9 21 D9 21 D9 22 B9 22 A2 22 B9
05C0  22 D1 22 D1 22 D1 21 F5 81 F5 20 00 22 D1 22 D1
05D0  21 F5 21 F5 22 D1 22 B9 22 D1 22 DD 22 DD 22 DD
05E0  22 81 C2 81 41 F5 41 D9 42 89 01 F5 41 F5 00 00
05F0  00 00

CARICARE INOLTRE I SEGUENTI VALORI
 00D2 F2
 00D3 05
```

Tabella II - Codice oggetto (per il microelaboratore AMICO 2000 del brano musicale "Love story" e relativo spartito.

Con questo programma di "Editing musicale" dunque i tasti del microelaboratore realizzeranno le seguenti funzioni:

[HLT] ha funzione di deviatore per 1) comporre, 2) eseguire un brano musicale. All'inizio si è in composizione. Premendo HLT viene eseguito il brano "editato". Premendo nuovamente HLT si torna in editing.

[RUN] ha funzione di deviatore per visualizzare 1) nota, ottava, durata (in sedicesimi), 2) il numero di battuta e il numero di nota della battuta relativi alla nota visualizzata. Esempio:

doH32

cioè Do≠ della 3ª ottava da 1 ottavo (2 sedicesimi).
Premendo RUN compare sul display ad esempio 0024 03, indicante che il Do≠ 3° ottava da 1 ottavo si trova nella 24esima battuta dall'inizio del brano musicale ed è la 3ª nota di questa battuta.

[0]...[F] Questi tasti hanno tre significati in sequenza rotativa, come segue:

1) [0]...[C] valgono rispettivamente secondo quanto riportato in Figura 7.

2) [1]...[6] identificano l'ottava come riportato in Figura 8.

3) [0]...[F] identificano la durata della nota in multipli di 1/16. Il tasto 0 equivale a 16/16.

4) Torna valere come il punto 1).
Se al punto 1 viene scelta una Pausa, la successiva impostazione identifica la durata della Pausa, poichè non ha senso determinare l'ottava.

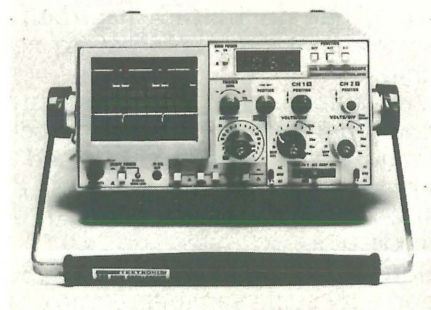
↑	inserisce la nota visualizzata nel brano musicale.
REG	cancella la nota visualizzata dal brano musicale.
DA	visualizza la nota precedente rispetto a quella attualmente visualizzata.
AD	visualizza la nota successiva rispetto a quella attualmente visualizzata.

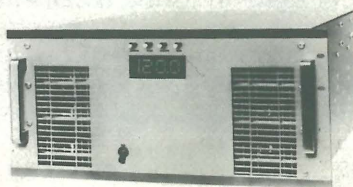
Conclusione
Ci è sembrato opportuno allegare a questo punto il testo di un brano musicale (v. Tabella II) introdotto nel microelaboratore con il programma precedentemente descritto. Nell'impossibilità dell'ascolto immediato, invitiamo i lettori volenterosi a verificare la correttezza della codifica.

STRUMENTAZIONE

Oscilloscopio con multimetro digitale

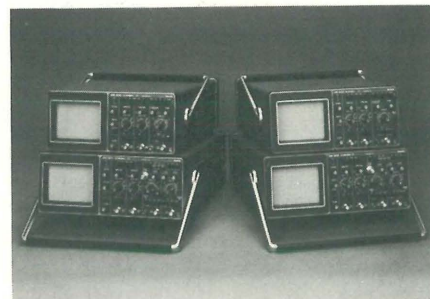
Il Tektronix 305 è uno strumento portatile per assistenza che combina in una sola unità un oscilloscopio ed un multimetro digitale, offrendo 5 MHz di banda passante, display a doppia traccia, alimentazione incorporata a batterie ricaricabili ed un multimetro digitale con tutte le funzioni. Eppure 305 è uno strumento di soli 4,8 Kg e di dimensioni estremamente compatte.
L'oscilloscopio multimetro mod. 305 è l'ideale per l'impiego nell'assistenza tecnica di controlli industriali, strumentazione sanitaria, centraline di relay telefonici, ecc.
Dotato di una cinghia per il trasporto che lascia libere entrambe le mani, il 305 può essere agevolmente portato su passerelle, scalette e in spazi ristretti normalmente impraticabili, grazie al suo peso ed ingombro contenuti.
In pratica il 305 costituisce lo strumento ideale per quasi tutte le applicazioni che richiedono un DMM e/o un oscilloscopio per le quali la portatilità e le piccole dimensioni costituiscono altrettanti vantaggi importanti ed il funzionamento a batteria addirittura una necessità.

SILVERSTAR - MILANO

Alimentatori senza interruzioni

La Elgar ha presentato un'altra serie di piccoli sistemi per fornire un'alimentazione senza interruzioni (UPS).
Il modello UPS - 102 - 1B, da 100 VA, è il primo prodotto della linea che usa uno switch a transistor al posto degli SCR nell'invertitore c.c./c.a. la sintesi e la regolazione dell'onda sinusoidale vengono effettuate con la modulazione della larghezza dell'impulso (PWM) usando una logica CMOS.
La nuova serie di MINI-UPS parte con potenze di targa di 250VA e sarà estesa fino ad unità da 5 KVA.
Tutti i modelli offrono una regolazione della tensione dell'1% con rendimenti che arrivano all'85% per potenze nominali più alte.

L'UPS - 102 - 1B è contenuto in uno chassis alto 17 cm, adatto per il montaggio in rack.
Il costo è di. 2295.

SISTREL - SESTO S. GIOVANNI

Oscilloscopi da 35 MHz

Gli oscilloscopi Philips PS 3216 e PM 3218 estendono le facilities già presenti nella serie da 25 MHz.
Il PM 3216 è un oscilloscopio a doppia traccia con base dei tempi

singola mentre il PM 3218 è un modello a doppia traccia con base dei tempi doppia.
Quest'ultimo è caratterizzato da una base dei tempi alternata che consente la visualizzazione simultanea sia del segnale principale che di quello ritardato.
La sensibilità per entrambi gli strumenti è 2 mV in tutta la larghezza di banda di 35 MHz, accoppita con una alta sensibiiità del trigger e una sensibilità del trigger esterno di 200 mV.
Inoltre un attenuatore con rapporto 10 : 1 sul trigger esterno fornisce un range dinamico estremamente ampio.
La massima velocità di sweep è 10 ns/div.
Una facility di trigger hold-off elimina il triggering doppio sui segnali digitali, così che non c'è la necessità di usare la base dei tempi nel modo non calibrato.
La modulazione dell'ingresso Z rende i nuovi oscilloscopi ideali per l'ulteriore aumento del numero delle applicazioni dell'analizzatore logico.
Anche questi strumenti sono stati studiati per essere il più facile possibile da usare. Per esempio entrambi sono realizzati col doppio isolamento, così che non è necessario il collegamento a terra, con la conseguente eliminazione dei loop verso terra e dei problemi di ronzio.

L'alimentazione può essere a 110, 127, 220, e 240 V c.a. (±10%) con frequenza da 46 a 440 Hz. È possibile anche il funzionamento in continua con tensione da 21 a 27 Vcc. Come opzioni si può avere una batteria interna a 24 V per avere uno strumento veramente portatile. Il consumo si aggira intorno ai 30 W e pesa solo 8,4 Kg.

PHILIPS - MONZA

Nuovo analizzatore di stati logici

La Paratronics Inc. presenta il nuovo analizzatore di stati logici a microprocessore Mod. 532. Esso permette l'analisi di 32 canali alla velocità di 12 milioni di parole/sec e ha una profondità di memoria di 250 parole.
Il display permette la visualizzazione dei dati in codice binario, ottale o esadecimale, mentre sono previsti 21 Modi diversi di trigger della sequenza di campionamento. Previsto per l'analisi di segnali TTL, DTL, RCL e ECL con l'uso di sonde dedicate, risponde in modo efficiente alle più svariate esigenze di ricerca manutenzione e produzione di apparati che usino microprocessori diversi.
È controllabile tramite terminale TTY o RS232, o ancora da linea telefonica per l'esecuzione di test a distanza. Ha la capacità di memorizzare fino a 8 programmi di test (dati, signatures associate e posizioni del pannello frontale) in un modulo RAM/EPROM e prevede delle procedure di confronto rapido fra dati del sistema sottoprova con quelli memorizzati. Collegabile con qualunque oscilloscopio o usabile anche da solo, presenta infine caratteristiche di compattezza e di leggerezza (5 Kg), tali da renderlo ideale nel servizio di manutenzione presso il cliente.

VIANELLO - MILANO

Misuratore di frequenza e periodo

Il 9520 della Racal è un misuratore di frequenza e periodo di impiego universale per una varietà di applicazioni nell'industria, nella ricerca e nelle istituzioni didattiche.
Il campo di frequenze d'ingresso si estende da 5 Hz a 10 MHz.
Il display di quattro cifre permette una lettura massima di 9,999 MHz, ma lo strumento può essere utilizzato per misurare dei segnali notevolmente superiori a qesta frequenza, funzionando fino a 20 MHz. A frequenze inferiori la risoluzione del display può essere migliorata misurando il periodo del segnale d'ingresso su uno dei quattro campi di periodo previsti. I tempi di clock ed il numero di periodi disponibili sono stati scelti in modo da fornire le letture più precise, compatibilmente con tempi di misura accettabili.
La sensibilità d'ingresso di 100 mV è conservata fino alla massima frequenza. Il circuito d'ingresso è protetto completamente contro i danneggiamenti involontari e sopporta dei sovraccarichi fino a 250 Vca efficaci o ±400 Vcc.
L'uso del 9520 è stato semplificato al massimo. Tutte le funzioni, compresa l'alimentazione principale, vengono selezionate mediante dei pulsanti di comando.
È possibile trasformare le letture visualizzate in forma BCD, adatta per il collegamento con registratori a nastro, stampanti, processori, rivelatori di superamento di limiti ed altre apparecchiature periferiche.
Il 9950 ha un assorbimento di potenza di 15 VA e può funzionare in un campo di temperature compreso fra 0°C e + 55 °C, con alimentazione a 100, 110, 200, 220 o 240 V e frequenza fra 45 e 440 Hz.

COMPONENTI

La XCARD: una scheda elettronica obliterabile per telefoni pubblici REAlizzata dalla SGS-ATES

La SGS-ATES, ha presentato la XCARD una nuova e rivoluzionaria scheda elettronica del formato di una carta di credito, da utilizzare nei telefoni pubblici a pagamento. Sviluppata in collaborazione con la S.I.P. (Società Italiana per l'esercizio Telefonico), la XCARD è dotata di una memoria non volatile di 136 celle di cui 100 utilizzate come strutture obliterabili. Le 36 celle rimanenti sono impiegate come codice di riconoscimento e controllo della scheda al momento dell'uso. Ogni tentativo fraudolento di cancellazione o rigenerazione della memoria andrebbe necessariamente a modificare tale codice, rendendo in tal modo la scheda inutilizzabile.
La memoria è costruita con una tecnologia originale sviluppata della SGS-ATES. Ogni cella è costituita da un singolo transistore MOS a canale - n, con una capacità di ritenzione di dati praticamente infinita.

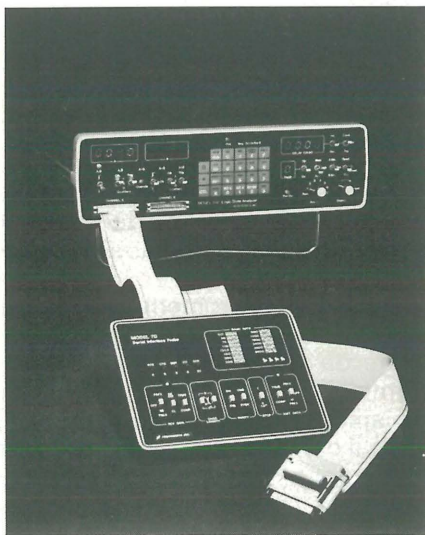
SGS - ATES AGRATE BRIANZA

La 2114 cambia tecnologia

La Intel ha riprogettato la classica memoria 2124 (RAM statica da 1 K * 4) con tecnologia HMOS. Ne è stato un prodotto su un'area di silicio inferiore del 32% rispetto alla 2114 e con caratteristiche molto interessanti per quanto riguarda sia la velocità che la potenza dissipata. La 2114 viene infatti fornita con le sigle 2114AL-2/3/4 e 2114A-4/5 con tempi di accesso che vanno da 120 A 250 ns e correnti di assorbimento che vanno da 40 a 70 mA.

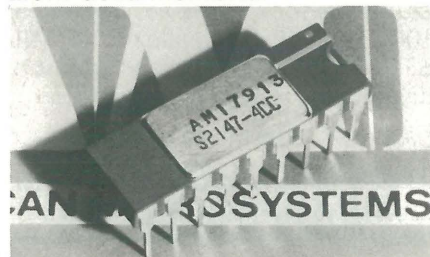
ELEDRA - MILANO

Interfaccia TV a microprocessore

L'M 106 della SGS-ATES è un dispositivo LSI programmabile per applicazioni controllate da un microprocessore nei campi TV e dei controlli industriali.
È realizzato con la tecnologia N-channel MOS Silicon gate, richiede una alimentazione singola a +5V e ha ingressi e uscite TTL compatibili.
Un'alimentazione a + 12V viene usata per polarizzare il circuito switch analogico incorporato nel chip.
L'interfaccia a microprocessore comprende un ingresso per clock monofase, un bus di sistema ad 8 bit bidirezionali, due ingressi di strobe e un'uscita di richiesta di interrupt.
Sono disponibili 7 segnali di uscita con duty cycle variabile. Dopo una

semplice filtrazione con filtro RC, questi segnali diventano le uscite analogiche del sistema.
Ci sono anche una uscita blanking a tre colori per visualizzare dati alfanumerici o grafici sullo schermo del tubo del televisione. Altre otto uscite digitali per scopi generali sono disponibili con la configurazione a collettore aperto.
Per i controlli analogici ci sono 6 D/A, con una risoluzione di 64 passi, mentre un convertitore D/A moltiplicatore a 13 bit (8192 passi) fornisce la tensione di sintonizzazione allo switch analogico.
la sezione del CRT display, basata su una matrica completamente programmabile 64 × 64, sotto controllo software, può lavorare con qualsiasi standard TV.
L'M106 viene fornito in un package dual-in-line plastico a 40 pin.

SGS-ATES - AGRATE BRIANZA

La AMI introduce la RAM statica VMOS ad alta velocità da 4K

La AMI Microsystems è entrata in produzione con una versione VMOS della 2147, la RAM statica ad alta velocità da 4K × 1.
La S2147-3 RAM VMOS, completamente statica, ha un tempo di accesso di 55ns, ormai standard industriale, mentre la S2147-5 ne ha uno di 70 ns.
Questo componente VMOS, completamente compatibile TTL, ha bisogno di una singola alimentazione da 5V, possiede internamente un circuito automatico di riduzione del consumo (power down) e non richiede clock nè cicli di rinfresco. Dato che il tempo di accesso dagli indirizzi e il tempo di chip enable sono uguali, il power down automatico non riduce la prestazioni della memoria.

La S2147 è fabbricata usando il processo VMOS brevettato dalla AMI, che permette di ridurre le dimensioni delle celle standard d RAM e di produrre in grosse quantità memorie ad elevate prestazioni.

AMI Microsystems - MILANO

Buffer/latch a 8 bit

La Motorola ha presentato un buffer/latch a 8 bit con uscita three-state.
Compatibile sia con la serie logica 74S che con i sistemi a microprocessori M6800, il dispositivo è contrassegnato con la duplice sigla MC6882/MC3482 e, per quanto riguarda la configurazione dei pin, è compatibile con il SN74S373.
Sono disponibili le versioni invertenti e non invertenti. Esse presentano il suffisso A e B, rispettivamente.
Il chip MC6882 è particolarmente adatto ad applicazioni bus oriented, quali la bufferizzazione del bus degli indirizzi tra il microprocessore M6800 ed i dispositivi di memoria o di controllo periferici, ad esso collegati.
Gli ingressi PBP ad alta impedenza rendono minimo il carico del bus e la presenza di isteresi su tutti gli ingressi rende più sicuro ed efficace l'immunità al rumore.
L'uso della tecnologia Schottky riduce il ritardo di propagazione tipico a 8 ns, e le uscite three-state permettono ai buffers non utilizzati di venire scollegati dal bus in modo efficace.
Il dispositivo funziona con una sola alimentazione +5V e lo stato di alta impedenza dell'uscita viene mantenuto sia con alimentazione che in sua assenza.
Gli ingressi LATCH e OUTPUT ENABLE sono bufferizzati in modo da costituire un solo conduttore logico.
Tutte le versione del MC6882 sono disponibili in un package dual-in-line a 20 pin, plastico o ceramico, per un funzionamento fra 0 e 75°C.

MOTOROLA - MILANO

Video timer/controller su chip singolo Texas Instruments

La Texas Instruments ha annunciato la disponibilità del nuovo video timer/controller su chip singolo: il TMS 9927.
Questo dispositivo realizzato in tecnologia N-CHANNEL SILICON GATE, ed assemblato in contenitore a 40 pin genera i segnali di temporizzazione video per monitor CRT standard e non standard sia con formati interallacciati che non interallacciati.
Il TMS 9927 può essere utilizzato con microprocessori ad 8 o 16 bit, inclusa la famiglia TMS 9900, come dispositivo di INPUT/OUTPUT selezionabili in memory mapping; inoltre può comunicare con l'interfaccia di CRU della famiglia 9900 attraverso il TMS 9901.
Il nuovo VIDEO TIMER/CONTROLLER comprende 5 sezioni:

* interfaccia per la CPU
* controllo di cursore
* controllo orizzontale
* controllo verticale
* autoinizializzazione

Esso fornisce 9 registri di controllo programmabili dall'utilizzatore; sette di questi controllano il formato sia orizzontale che verticale, gli altri due controllano la posizione del cursore.
L'architettura di TMS 9927 è studiata per fornire la massima flessibilità di progetto. La maggior parte dei CRT-RASTER SCAN può essere controllata dal TMS 9927 semplicemente programmando in maniera opportuna i registri di controllo.
TMS 9927 è intercambiabile con il CRT 5027 della STANDARD MICROSYSTEMS CORPORATION (SMC).

CRAMER - ROMA

SBC 941 Industrial Digital Processor

Il dispositivo SBC 941 è un componente integrato a 40 pin particolarmente adatto per un'ampia gamma di applicazioni nell'ambito dei controlli industriali di tipo digitale.
Opera sostanzialmente come dispositivo slave sotto il controllo di un CPU master che gli richiede di eseguire una delle nove funzioni primarie che sono programmate nel suo interno.
Ogni funzione primaria esplica una specifica azione I/O; queste funzioni sono:

EVENT	: controllo e conteggio di eventi su 8 input diversi.
FCOUNT	: misure di frequenza su 8 input diversi.
FREQ	: generazione di frequenze su 8 output diversi.
PERIOD	: misure di periodo su 4 input diversi.
SCAN	: controllo della variazione di stato su 16 input diversi.
SERIN	: ricezione di dati secondo lo standard di comunicazione seriale asincrono.

SEROUT : trasmissione di dati secondo lo standard di comunicazione seriale asincrono.
SHOT 1 : generazione di impulsi con ritardo e periodo variabile su 8 output diversi.
STEPPER : generazione di 8 output programmabili per il controllo di stepping motor.

Quale dispositivo slave si interfaccia direttamente, come un normale componente periferico, alla struttura di bus dei sistemi MCS 80/85/86, svincolando totalmente la CPU con cui lavora da ogni compito di elaborazione di I/O di tipo digitale.
La sua disposizione dei piedini è completamente compatibile con quella dello 8041A Universal Peripheral Interface (UPI-41A) per cui può essere normalmente usato sulla piastra SBC 569 Intelligent Digital Controller quale controller di I/O con notevole risparmio di tempo e di fatica di programmazione da parte dell'utilizzatore.
Il componente SBC 941 richiede la sola alimentazione +5 V e tutte le sue linee di I/O sono TTL compatibili.

ELEDRA - MILANO

Nuovo UART della famiglia 6500

La Rockwell e la Synertek hanno annunciato la disponibilità sul mercato italiano del componente con sigla R6551 e SY6551 rispettivamente.
Esso funziona come adattore asincrono di interfaccia per comunicazione rivolto a sistemi che usano microprocessori 6500 o 6800.
Incluso nello stesso integrato è il generatore di velocità, che richiede solo il montaggio esterno di un cristallo di quarzo.
Sono programmabili a software 15 diverse velocità di trasmissione/ricezione che vanno da 50 a 19200 baud. Gli UART comunicano con il microprocessore attraverso il bus dati, quattro segnali di selezione indirizzo e tre segnali di controllo, mentre si collegano al moden o al terminale, usando dei traslatori di livello in RS 232.
La trasmissione e ricezione dati possono operare a velocità fornibili dall'esterno e indipendenti tra di loro, mentre la disponibilità dei caratteri può essere segnalata sia ad interrupt che ad analisi dello stato.
Sono programmabili a software inoltre la lunghezza della parola (da 5 a 9 bits), il numero dei bits di stop, la generazione del bit di parità e relativo controllo insieme a segnali di controllo di interfaccia ai moden e ai data-set (DCD e DSR). Possono inoltre essere gestite sia operazioni in half-duplex che in full duplex.

ROCKWELL DE MICO - MILANO
SYNERTEK COMPREL - CINISELLO B.

La nuova Z80-DART della Zilog gestisce due canali asincroni full-duplex

La Zilog ha recentemente ampliato la famiglia Z80 introducendo la Z80-DART (Dual Asinchronous Receiver Transmitter), un componente periferico molto versatile che mette a disposizione due canali indipendenti full-dotati di due insiemi distinti di linee di controllo per moden.
La Z80-DART è in grado di soddisfare un vasto insieme di esigenze nel campo della comunicazione asincrona seriale, specie nell'area di impiego dei microcalcolatori
Usata come convertitore serie-parallelo e parallelo-serie può essere impiegata in terminali intelligenti e no, e come controllore in ogni apparato che comunichi solo attraverso protocolli asincroni.
Le caratteristiche principali della Z80-DART sono:
- controlli per modem separati ed indipendenti per ciascun canale, con capacità di "monitorare" lo stato del modem
- ricezione dei dati attraverso quattro registri di buffer trasmissione attraverso doppio buffer
- vettore di interruzione programmabile; il vettore di interruzione inoltre è alternabile automaticamente a seconda delle condizioni che causano l'interruzione (status affects vector), permettendo così una più rapida risposta
- priorità attraverso "daisy-chain" secondo lo standard Z80, che non richiede logica esterna per la vettorizzazione e la determinazione delle priorità delle periferiche
- frequenza di comunicazione da 0 a 500 Kbit/sec con clock a 2.5 MHz, da 0 a 800 Kbit/sec con clock a 4 MHz
- opzioni di programmazione: 1, 1 e 1/2, o 2 bit di stop; parità pari, dispari, o abolizione del controllo di parità; frequenza di clock x1, x16, x32 e x64
- riconoscimento e generazione delle condizioni di "break"
- riconoscimento degli errori di parità, di traboccamento del buffer di ricezione (OVERRUN) e di errore di "frame".

La Z80-DART è disponibile in package ceramico o plastico a 40 piedini "dual in line", e intervallo di temperatura normale, esteso o militare.

ZELCO - MILANO

Componenti programmabili con uscite 3-state

Am25LS2568 e Am25LS2569, contatori 3-state in tecnologia Low Power Schottky sono contatori programmabili up/down rispettivamente BCD e binario. Essi presentano un ritardo massimo clock-uscite di 27 nsec.
L'uscita di "ripple carry" permette la connessione in cascata di vari dispositivi mentre quella di "clock carry" permette di avere a disposizione un clock di pilotaggio libero da "spifferi".
Questi dispositivi prodotti dalla ADVANCED MICRO DEVICES sostituiscono i cinque SSI ed MSI finora impiegati per realizzare la stessa funzione.
Essi sono disponibili nel nuovo contenitore standard a 20 piedini sia plastico che ceramico nonché nel contenitore flat-pack particolarmente raccomandato per applicazioni militari.
Questi dispositivi sono testati al 100% in conformità alle norme MIL STD-883 classe C senza variazioni di prezzo.

CRAMER - ROMA

MICROPROCESSORI

La AMI annuncia un microprocessore 6800 più veloce

La AMI Microsystems dispone di una versione del microprocessore 6800 in grado di eseguire un'istruzione in 800 ns, tempo che è inferiore di 200 ns rispetto a qualsiasi

altri 6800 normalmente sul mercato.
L'S68H00 funziona con un clock da 2.5 MHz e può eseguire una istruzione di accesso dati in due cicli di clock, pur rimanendo identico sotto il punto di vista funzionale alle altre versioni di 6800 più lente. Con una velocità di lavoro superiore di 2.5 volte quella normale, questo componente è particolarmente indicato per applicazioni quali trattamento ad alta velocità di segnali nella strumentazione, controlli numerici, ecc.
Come i vecchi 6800, l'S68H00 tratta 8 bit in parallelo, ha un bus dati bidirezionale da 8 bit e un bus indirizzi da 16 bit, con un indirizzamento di 65.536 bytes. Il suo set comprende 72 istruzioni, con sette sistemi di indirizzamento, e un tempo di esecuzione che va dai 2 cicli macchina per un software interrupt.
È stata anche annunciata l'S68H21, una versione più veloce della PIA, che opera anch'essa a 2.5 MHz.
Questo componente, insieme alla RAM AMI S2114 (1k x 4) e alla ROM S4216B (2k x 8), permette di implementare un sistema di microcomputer ad alta velocità.

AMI MICROSYSTEMS - MILANO

Prototipi di Z8 ottenibili dalla Zilog

La Zilog ha annunciato la disponibilità dello Z8-02 MPD, una nuova versione a 64 pin del suo chip di microcalcolatore a basso costo ed alte prestazioni.
Il Z8-02 MPD è inteso in paticolare per la parte di sviluppo di prototipi, nella quale ci si aspetta dei cambiamenti ai programmi man mano che lo sviluppo procede. La versione a 64 pin è una variante, sprovvista di ROM, dello Z8 standard a 40 pin. Lo Z8-02 MPD fornisce all'esterno 12 indirizzi per la ROM e le 8 linee di dati e di controllo relative. Per lo sviluppo dei 2KByte di programma (che dovrebbero risiedere poi nella ROM) possono essere usate delle EPROM, PROM o RAM. Quando poi il programma è messo a punto esso verrà in genere trasferito su ROM con opportuna mascheratura nel caso di applicazioni che richiedano larghi volumi. (Per piccoli volumi v'è un'altra interessante possibilità che verrà tra poco annunciata).
Il nuovo Z8-02 MPD è anche utilizzabile in applicazioni richiedenti piccole quantità, per le quali la mascheratura delle ROM è proibitiva e dove è invece accettabile l'uso di una PROM esterna.
Tutte le versioni dello Z8 offrono grande adattabilità ai bisogni dell'utente, grande velocità, utilizzazione efficiente della memoria, possibilità sofisticate di interrupt e di manipolazione di bit etc. La versione standard dello Z8 offre 2 KByte di ROM, 144 Byte di RAM (utilizzabili anche come registri), 32 linee I/O, due contatori temporizzatori e porte seriali.
Campioni di Z8-02 MPD sono ottenibili da qualche tempo in un package ceramico "quad-in-line".

ZELCO - MILANO

Il Protopack Z8 della Zilog consente un economico sviluppo dei programmi

Una tecnica unica di impaccamento che semplifica lo sviluppo dei programmi durante lo stadio di produzione del prototipo permettendone una economica memorizzazione su una EPROM separata, è offerta in una nuova versione del single-chip Z8 della Zilog.
Lo Z8-03 MPE (microcomputer Protopack-Emulator), cioé la versione standard dello Z8 senza ROM, è l'ideale per lo sviluppo dei prototipi e la produzione di piccole serie iniziali prima di passare a quelle impieganti lo Z8 standard con ROM. Impaccato in un contenitore con 40 pin che lo rende compatibile ed intercambiabile a livello di pin con lo Z8 standard, il Protopack porta "in groppa" uno zoccolo a 24 pin su cui si inserisce direttamente la EPROM contenente il programma. Questo zoccolo è provvisto di 12 linee per indirizzare la ROM (in questo caso EPROM), 8 linee per i dati e le necessarie linee di controllo per l'uso con le EPROMs 2716.
La compatibilità dei pin permette all'utilizzatore di progettare un circuito stampato fatto per lo Z8 a 40 pin, programmabile con mascheratura delle ROMs, mentre può invece utilizzare inizialmente un protopack Z8 per la costruzione dei prototipi e della produzione pilota. Quando si ottiene il programma definitivo, l'utilizzaore può ritornare allo Z8 standard per produzioni di grosso volume senza cambiare la piastra.
In applicazioni con quantità ridotte, il Protopack Z8 consente all'utilizzatore di evitare i lunghi tempi di attesa per la mascheratura del programma, e gli alti costi relativi, ottenendo invece una maggiore flessibilità di programmazione.
A differenza della versione convenzionale con EPROM dei microomputer a single-chip, che avevano finora una EPROM intera, il Protopack Z8 permette la memorizzazione del programma in una EPROM separata e di tipo standard. Ciò porta ad un economico sviluppo di programmi in applicazioni in cui è utilizzata la stessa configurazione hardware per più di un programma. Inoltre la EPROM è facile da programmare o da sostituire essendo di tipo standard.
Come lo Z8 standard e lo Z8-02 MPD a 64 pin (microcomputer prototipo in single-chip), il Protopack Z8 offre un'esecuzione delle istruzioni ad alta velocità, uso efficiente della memoria, possibilità sofisticate di interrupt, di manipolazione di testi ed espandibilità del sistema. Il componente opera con una sola alimentazione a +5 Volt. Lo Z8 standard a 40 pin offre 2K x 8 ROM, 144 x 8 RAM, 32 linee I/O, 2 contatori/temporizzatori ed una porta seriale asincrona.
Sono disponibili campioni dello Z8-03 MPE in package a 40 pin dual-in-line a $ 110 in quantità da 10 a 99.

ZELCO - MILANO

Microelaboraori a singolo chip

Appartengono ad una serie della National, formata da sistemi completamente programmabili, progettati per l'esecuzione di operazioni di calcolo e di controllo dell'I/O, con una velocità di 400.000 operazioni al secondo. Sono:
- l'INS8048, che può operare come sistema autonomo a chip singolo, con programmi memorizzati in 1 Kbyte (8 K bit) di memoria ROM mascherata e 64 byte di memoria RAM,
- l'INS8049, con 2 K byte (16 K bit) di memoria ROM masked e 128 byte di memoria RAM,
- l'INS8050, microelaboratore a singolo chip, ad alta densità con 4 K byte (32 K bit) di memoria ROM masked, pin-to-pin compatibile con i modelli 8048/8049 e 256 byte di memoria RAM,
- gli INS8038, INS8039, INS8040

versioni senza memoria ROM degli INS8048, INS8049 e INS8050.

Tutti i sei dispositivi contengono gli stessi sistemi di base di controllo e di elaborazione in un package compatibile a 40 pin: un processore centrale general-purpose parallelismo 8, stack array estendibile, 96 istruzioni e un ciclo di 2,5 μs per la versione 6 MHz a 1,36 μs per la versione 11 MHz; tre porte di I/O programmabili e otto linee di temporizzazione contatori programmabili per temporizzazioni di eventi; controlli di interruzioni prioritarie; generatore del clock del sistema.

L'uso del processo XMOS si traduce in numerosi vantaggi. Nel INS8049 la misura del chip è solo 185 per 185 mA.

Il consumo del INS8049 è di 605 mW, cioé quasi dal 20 al 25% in meno di quello richiesto sui dispositivi attuali che usano processi standard tipo NMOS.

Sebbene tutti i dispositivi della serie siano progettati come dirette sostituzioni pin-to-pin e funzione-per-funzione dei dispositivi della serie INTEL MCS-48, la serie incorpora un numero di miglioramenti nelle prestazioni del chip.

Nell'INS-8049, per esempio, la corrente nella condizione stand-by per la memoria RAM è programmabile a seconda della quantità di dati in memoria richiesti.

La tensione nella condizione standby è di soli 2,2 V.

NATIONAL SEMICONDUCTOR - MI-

MC 6809 il ponte tra 8 e 16 bit

MC 809 è la seconda generazione Motorola di microprocessori ad 8 bit che non solo rappresentano un miglioramento nei confronti di 6800, ma realizzano un ponte tra la famiglia ad 8 bit 6800 e la famiglia a 16 bit 68000. Le innovazioni nelle caratteristiche sia SOFTWARE che HARDWARE inserite in 6809 sono tali da renderlo il microprocessore NMOS ad 8 bit più potente del mercato.

MC 6809 fu specificatamente progettato per supportare e favorire l'uso di metodi di programmazione e strutturato con gli obiettivi di ridurre i costi di sviluppo e manutenzione del Software.

L'architettura di 6809 include nuovi registri, nuove istruzioni, nuovi modi di indirizzamento e nuove caratteristiche hardware che ne aumentano la perfomance da 2,5 a 5 volte rispetto ad MC 6800.

E cioé:

* Bus dati ed indirizzi compatibili con la famiglia M6800 permettono l'interfacciamento con tutte le periferiche standard 6800;
* Bus clock sino a 2 MHz, con possibilità di generatore di clock interno o con pilotaggio di clock esterno per la sincronizzazione di sistemi a multiprocessore.
* Segnali di controllo aggiuntivi: BUSY, LAST INSRUCTION CYCLE, MEMORY, READY BUS REQUEST oltre a segnali di SYNC ACK ed INTERRUPT ACK per la modifica esterna del vettore interrupt.
* I livelli di interrupt aumentati con l'aggiunta di 2 livelli Software ed un livello Hardware: FAST IRQ a risposta veloce.

Il potenziamento dell'architettura interna del Software corrispondente sono le caratteristiche più imortanti di 6809.

* Software sorgente upword compatibile con 6800.
* Set di registri ampliato comprendente tutti i registri di 6800 più un ulteriore registro indice a 16 bit Y, uno user stack pointe U, un accumulatore D (ottenuto concatenando gli accumulatori A e B) ed un registro pagina DP per il posizionamento della pagina zero in memoria.
* 10 modi di indirizzamento, oltre quelli già presenti con 6800, il long relative con offset a 16 bit e l'indiretto sia indexed che esteso.
* Potente indirizzamento indexed; cinque registri utilizzabili come registri indice X, Y, S, U e PC con varie opzioni per la definizione dell'offset: Zero, zero con autoincremento o decremento di 1 o 2, con segno a 5, 8 o 16 bit, con accumulatori A, B o D.
* Istruzioni per operazioni su 16 bit.
* Istruzioni addizionali per sincronizzazione di interrupt con eventi esterni. Va notato come l'architettura ed i modi di indirizzamento di 6809 supportino completamente tecniche di programmazione strutturate, sottoprogrammi restanti e segmenti di programma modulari che possono essere resi indipendenti dalla loro posizione in memoria. Questa caratteristica rende possbile la progettazione di package software ROM residenti.

Motorola prevede due tipi di package per lo sviluppo di circuiti utilizzanti MC 6809:

1) **Advance support**
M6809 MASCO 010M cross macro assembler
M6809 SIM simulatore
entrambi per sistemi Exorciser/Exoterm
2) **Support products**
MEX 6809-1 (MEX 6809-2) modulo si supporto completo MPU/USE con firmware (EX-BUG) e SOFTWARE per aggiornare sistemi EXORCISER/EXORTERM a M6809 EXORCISER.

Il supporto che verrà sviluppato futuro prevede anche software ad alto livello quali PASCAL, MPL, FORTRAN, COBOL, BASIC e REAL TIME OPERATING SYSTEM.

CRAMER - ROMA

SISTEMI DI SVILUPPO

Sistema di sviluppo GenRad

Il sistema GenRad Futuredata 2300 è un sistema di sviluppo che, in accordo alle nuove esigenze progettuali e di mercato, ha un'architettura a più stazioni di lavoro. Questa soluzione permette di ridurre il costo per utente, data la possibilità di suddividere i costi di dispositivi periferici costosi quali stampanti e dischi. Il numero massimo di utenti in grado di lavorare contemporaneamente è 8. Ciascun posto di lavoro consiste di una stazione 2300 dotata di propria CPU, video e tastiera, ed in più configurabile in modo adatto mediante moduli di emulazione relativi a vari microprocessori. Inoltre vi è la possibilità di attuare le funzioni di una analizzatore di stati logici.

I microprocessori attualmente supportati sono l'8080, l'8085 e l'8086 della Intel; il 6800 e il 6802 della Motorola; lo Z80 della Zilog. È prevista l'estensione anche allo Z800.

Per quanto concerne lo sviluppo dei programmi, il sistema GenRad Futuredata 2300, oltre a fornire le tipiche funzioni di editing, assemblaggio e linkaggio, offre mezzi di sviluppo notevoli quali macroassemblatori e compilatori per BASIC e Pascal, di modo che i tempi di sviluppo di un programma vengono notevolmente ridotti.

GENRAD - MILANO

Emulatore per il prototipaggio dei microprocessori

La Mostek ha introdotto un sistema di valutazione, prototipaggio e test portatile a basso costo per i microprocessori single chip 3870, 3872, 3874, 3876.
Il sistema EVAL 70, composto di una singola scheda, incorpora un monitor/debugger in ROM (DDT-70) che consente l'esecuzione passo-passo dei programmi, l'inserimento di break point, la modifica di celle di memoria di porte di I/O, la programmazione di memorie EPROM da 1,2 e 4K bytes.
L'operatore interagisce con l'EVAL-70 tramite una tastiera funzionale e display a 7 segmenti.
L'EVAL-70 dispone di memoria RAM da 1K byte espandibile a 4K e di zoccoli per 4K di memoria EPROM.
Il sistema è inoltre dotato di cavo di emulazione con zoccolo da installare sulla scheda dell'utente.
Come opzione è disponibile un'interfaccia seriale per trasfeire programmi dal sistema di sviluppo MOSTEK SYS-80FT.

MOSTEK ITALIA - MILANO.

Nuovo sistema di sviluppo a disco rigido Intel

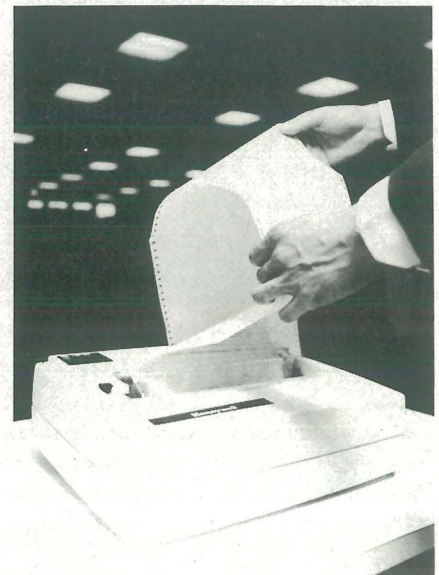
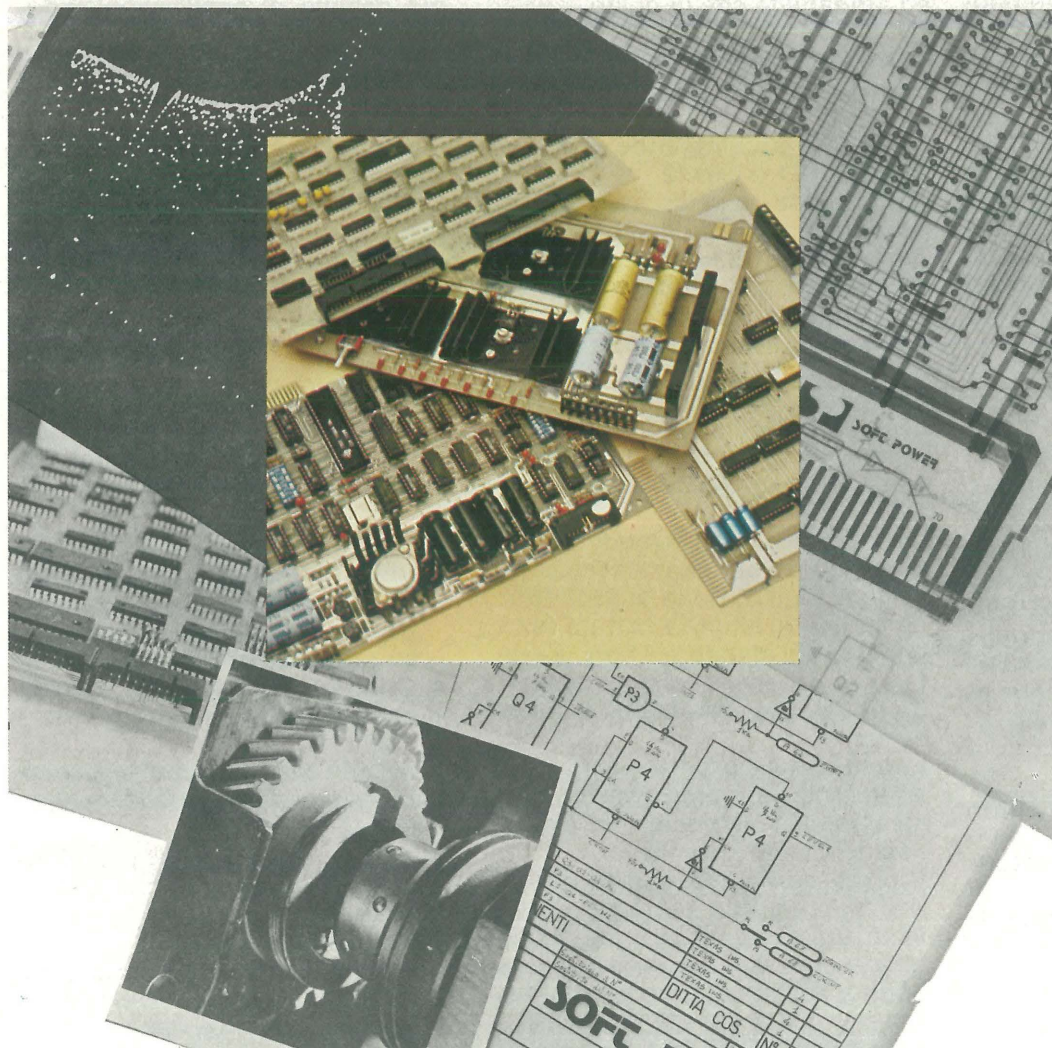
Mediante un'unità a disco rigido e una procedura per l'introduzione di sequenze complesse di comandi, l'ultimissima versione del sistema di sviluppo Intellec facilità l'implementazione dei programmi.
Il Modello 241 dell'Intel consente la compilazione o l'assemblaggio a velocità molto alte.
La supervelocità e la capacità dell'unità di memoria di massa a disco rigido sono ulteriormente rafforzate da un sistema operativo e raddoppiate prestazioni che è pure compatibile con tutto il software esistente.
Il disco rigido ha una latenza tipica di 50msec contro i 340 msec contro i 340 dei floppy disk. L'unità suddetta contiene due dischi, ognuno dei quali può memorizzare 3,65 Mbyte di dati e trasferirli alla velocità di 2,5 Mbit/sec. Uno dei due dischi è una cartuccia intercambiabile tipo IBM-5440 che permette all'utilizzatore di riconfigurare in pochi minuti il Modello 241 secondo le proprie esigenze software.
Poiché le cartucce possono essere scambiate. l'unità a disco rigido consente di fatto una memorizzazione illimitata di file di listing, di file di programmi-oggetto linkati e allocati e di altri dati.
In tal modo l'intercambiabilità dei supporti a disco rigido consente a numerosi utenti di condividere le prestazioni del Modello 241.
L'altro disco è fisso, non intercambiabile, e permette di ottenere un elevato rendimento per il software di sistema e per i programmi di utilità condivisi da tutti gli utilizzatori, ad esempio per i sistemi operativi, i compilatori, i programmi di "link" e "locate" e le librerie di sottoprogrammi.
Il più elevato "bit rate" dell'unità a disco rigido (2,5 MHz contro 0,5 MHz del floppy disk) richiede uno speciale controllore progettato utilizzando i componenti a microcomputer bipolare "bit-slice" della serie 3000 della Intel. Il controllore contiene 1 Kbyte di memoria RAM ad alta velocità con funzione di buffer per minimizzare l'occupazione del bus.
Oltre all'unità a disco rigido, il Modello 241 contiene un'unità per un floppy disk a singola densità, incorporata nella struttura principale del sistema insieme alla tastiera ed al CRT. Il floppy disk permette veloci trasferimenti di file di programmi sorgente e oggetto entro il sistema oltre ad archiviare piccoli file di dati.
La nuova versione 4,0 del sistema operativo ISIS-II, combinata con l'unità a disco rigido, raddoppia le prestazioni del Modello 241 rispetto ai normali sistemi a floppy disc.
La versione 4,0 dell'ISIS-II è totalmente compatibile con il software sviluppato dalla Intel e dagli utenti per le precedenti versioni del sistema operativo. Occupa il medesimo spazio di memoria di 12 Kbyte assegnato alle precedenti versioni: tale spazio contiene anche i programmi di gestione dell'unità a disco rigido. La versione 4,0 dell'ISIS-II può funzionare anche con i sistemi Intellec Modello 221, 231, MDS-800 della Intel, se dotati di floppy disk.
Tuttavia, con il Modello 241 a disco rigido l'utente può usufruire delle nuove possibilità di manipolazione dei file.

Ogni singolo disco rigido può contenere fino a 992 file contro i 200 del floppy disk. La possibilità di BAD-SPOT consente un buon recupero dell'errore nei supporti a disco magnetico e minimizza la probabilità di errore controllando tali supporti nella fase inizializzazione.

ELEDRA - MILANO

PERIFERICHE

S10: la nuova stampante seriale della Honeywell destinata al settore del "personal computing"

È questa la più piccola della serie di stampanti seriali a matrice progettata e prodotta dalla Honeywell Information Systems Italia, una serie che comprende ormai una ventina di modelli culminante nella teleprinter R 28 presentata all'inizio dell'estate. La S10 ha una velocità di 80 caratteri al secondo (con stampa bidirezionale), 80 posizioni di stampa, è dotata di una testina di stampa di alta efficienza e di un'elettronica di controllo estremamente evoluta e integrata. L'interfaccia è EIA-RS232C seriale asincrona di piccole dimensioni e di costo contenuto, essa è destinata all'impiego sia come hard copy (cioé come stampante affiancata a un videoterminale) sia come commponente di stampa nei piccoli e piccolissimi sistemi. In particolare si prevede che essa avrà una grande diffusione nelle applicazioni evolute di "personal computing", un settore. com'è noto, in pieno boom.
A conferma delle sue grandi possibilità di diffusione sta il fatto che già fin d'ora, ad annuncio appena avvenuto e prima ancora dell'en-

I problemi che un sistema di controllo deve risolvere in ambiente industriale sono tali da richiedere una progettazione orientata a questo scopo, non un adattamento di schede "general purpose". Dall'esperienza SOFT POWER:

Il sistema modulare a µP per l'automazione industriale SP/99

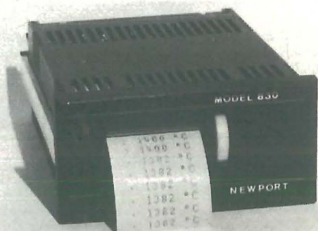
Elementi base del sistema sono i microcomputers a 16 e 8 bit che utilizzano i µP Texas Instruments TMS 9900 e 9980-81.
La potenza del microprocessore utilizzato permette la realizzazione di sistemi dalle elevatissime prestazioni e l'adozione di "soluzioni software" anche per problemi assai complessi e critici.
La famiglia di moduli SP/99 fornisce lo strumento adatto a risolvere i più diversi problemi di automazione. Ciascuno dei moduli del sistema SP/99 svolge una o più delle funzioni tipiche dei sistemi industriali:

- conversione analogica/digitale e digitale/analogica;
- acquisizione di informazioni di posizione da encoder incrementali;
- interfaccia per segnali in ingresso disaccoppiati otticamente e per segnali in uscita a 110/220 V AC;
- gestione di comunicazione uomo/macchina con impostazione da tastiera e visualizzazione su schermo CRT o display alfanumerici;
- comunicazione seriale sincrona/asincrona.

trata in produzione di questo nuovo modello, la Honeywell Information Systems Italia ha ricevuto per esso ordini per parecchie migliaia di esemplari. In particolare sono stati annunciati un ordine della Wenger Datentechik di Basilea per 2.000 esemplari e un ordine della Mistral S.p.A. di Latina Scalo per 2.400. Distibutori della S10 risultano a tutt'oggi la Segi di Milano e la Microlem Data di Vimercate.

HONEYWELL - I.S.I. - MILANO

Stampante a microprocessore

La data printer modello 830 della Newport Electronics misura solo 48 × 96 mm (pannello standard DIN) e ha una profondità di 140 mm. Possono essere stampate fino ad 8 colonne di dati su carta termosensibile di tipo commerciale, larga 35 mm. La stampante accetta in ingresso dati in BCD parallelo, in BCD seriale o in forma asincrona (RS232-C).
Il controller della velocità di stampa incorporato può essere usato nel modo seriale, il numeratore (contatore di sequenze) può essere usato sia in modo parallelo che seriale.

Il meccanismo di pilotaggio è caratterizzato da due sole parti in movimento (un motore a passi e l'albero di drive), in modo da avere un funzionamento silenzioso e senza manutenzione.
Nonostante le piccole dimensioni, il modello 830 è dotato di un alimentatore incorporato a 220 Vca/50 Hz.

RADIEL - SEGRATE

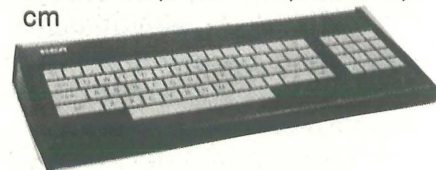
Nuovo drive a disco fisso

La BASF ha annunciato l'imminente uscita sul merato di drives a disco fisso della serie 6170. Sfruttando la tecnologia Winchester 3350 associata ad innovazioni di progetto esclusivo BASF intensamente collaudate, ne è uscita una periferica di memorizzazione dati altamente efficiente sia in termini di capacità ed affidabilità che di costo.
I 6170 presentano dimensioni uguali a quelle di un floppy disc drive, tranne la lunghezza, che è maggiore di 10 cm.
La serie ora comprende due modelli: il 6171 e il 6172.
Il primo usa un disco fisso ad una facciata, il secondo due dischi per tre facciate, con capacità complessiva di 8 megabytes per il primo e 24 megabytes per il secondo.
Il tempo di accesso medio è di 42 millisecondi.
Il supporto usato è il disco Winchester di 210mm di diametro che viene fatto ruotare da un motore a corrente continua senza spazzole a 3600 giri/m.
La codifica usata è la modulazione di frequenza per una densità massima di registrazione di 6542 BPI.
Le testine di lettura/scrittura sono poste su un attuatore a bobina mobile con caratteristiche molto buone di velocità, affidabilità e precisione.
La scelta del disco fisso inoltre elimina tutti quei problemi di intercambiabilità provocati dal disallineamento della testina.
Il disco con la meccanica dell'attuatore è sigillato a prova di polvere in un contenitore a coperchio trasparente. Sono esterni invece la bobina dell'attuatore e il motore di rotazione, il che facilita sia la dissipazione di calore che l'eventuale sostituzione degli stessi.
Sono disponibili da parte della stessa BASF per il 6171 e il 6172 tre configurazioni di interfaccia:
- interfaccia BASF SMD, che è quella tipica dei floppy disk drive in versione oem
- interfaccia bus BASF previsto per un collegamento economico ad un sistema di porte I/O con mini-/microprocessore
- interfaccia BASF e controller/formatter che permette il collegamento diretto a microcomputer, sollevando da qualunque problema di controllo diretto e formattazione.

MICROLEMDATA - VIMERCATE

Tastiere ASCII complete a basso costo

La nuova serie VP-600 della RCA rende disponibile a costo veramente basso 2 diverse tastiere di tipo professionale, complete di «encoder» per l'intero set ASCII a 128 caratteri alfanumerici, utilizzabili in un'ampia gamma di applicazioni industriali e consumer.
Entrambe utilizzano la moderna tecnologia a membrana flessibile per realizzare i vari tasti, i cui contatti risultano implicitamente protetti dall'umidità e dalla polvere.
L'efficienza del contatto è garantita per più di 5 milioni di operazioni, ed un generatore di toni incluso sulla piastra della tastiera genera automaticamente un feed-back di tipo acustico alla pressione di ogni tasto.
La tastiera per funzionare necessita della sola tensione di alimentazione + 5 Volt ed include un LED indicatore del «power-on».

Specifiche
- Set di caratteri: 128 caratteri ASCII (102 caratteri con i soli caratteri alfanumerici maiuscoli su selezione dell'utente)
- Codifica: 7 bit ASCII in parallelo, bufferizzati
- Tastiere:
  VP-601 58 tasti con layout di macchina da scrivere (2 tasti definibili dall'utente)
  VP-611 come VP-601 più 16 tasti per la tastiera solo numerica separata
- Rolover: 2 Key
- Segnali di Handshake: KD (Key Down), KD. Il segnale permane finchè il tasto non viene rilasciato
  Strobe, Strobe. Impulso di 15 msec nominali dopo la pressione del tasto
- Bit di parità: Pari, non bufferizzato
- Feedback acustico: 1 KHz nominale.
  Altoparlante a 8 Ohm con controllo del volume d'uscita
- Uscite: CMOS e TTL compatibili, ogni uscita può pilotare 2 carichi TTL
- Dimensioni:
  VP-601 33,3 cm x 17,8 cm x 5,1 cm
  VP-611 41,9 cm x 17,8 cm x 5,1 cm

- Peso:
  VP-601 1.360 gr
  VP-611 1.590 gr
- Alimentazione: + 5 V DC, 85 mA nominali LED indicatore di «power-on».

ELEDRA - MILANO

Stampanti di linea per applicazioni commerciali

Si tratta di 2 stampanti messe a punto dalla Radio Shack per l'impiego con i suoi sistemi a microcomputer TRS-80, adatti anche per la maggior parte degli altri computer che hanno porte di I/O parallele.
La TRS-80 Line printer III fornisce linee di 15 pollici con 132 caratteri e stampa, ad una velocità di 120 car/s con una matrice 9 x 7 punti, lettere maiuscole e minuscole. la testina stampante stampa in entrambe le direzioni mentre si muove attraverso la carta.
È possibile il posizionamento di forme prestampate quale assegni, comandando incrementi di 1/8 di linea.
È possibile mediante software scegliere caratteri espansi per avere intestazioni o parole evidenziate nel testo stampato. La Line Printer III è anche caratterizzata dalla funzione di autotest. Una circuiteria a microprocessore incorporata controlla la stampante. Le istruzioni di funzionamento di base e le routine di autotest sono memorizzate in una ROM. Anche la TRS-80 Line Printer II fornisce lettere espanse, sia amaiuscole che minuscole, usando un formato di matrice a 7 x 7 punti.

La sua velocità è di 50 car/s su linee di 8 pollici di 80 caratteri ciascuna.
La Line Printer II funziona sia con alimentazione della carta a frizione che a pin e può usare rotoli o fogli singoli.

RADIO SHACK - Milano

PERSONAL COMPUTERS

Anche in Italia il personal computer Apple II Plus

Da alcuni mesi è disponibile anche sul mercato italiano l'Apple II Plus, un personal computer tra i più versatili esistenti sul mercato, e che quindi trova applicazione in moltissimi campi, dallo small business all'uso professionale ed hobbistico. L'Apple II viene usualmente considerata la macchina più a basso costo tra quelle utilizzabili a livello professionale, e ne è prova il fatto che negli USA ne sono state vendute più di 100.000.
È programmabile in molti linguaggi: BASIC esteso Applesoft, Pascal, Assembler, Microlisp, Forth e Pilot. È inoltre dotato di due diversi sistemi operativi per la gestione di informazioni su diskette: DOS per i linguaggi BASIC, UCSD per il linguaggio Pascal e l'Assembler.
Internamente, il computer è costituito da una sola piastra su cui trovano posto tutti i componenti:

- CPU con microprocessore 6502 della Mos Technology
- 12 K bytes di memoria ROM, contenenti il Monitor e l'interprete BASIC
- da 16 a 48 K bytes di memoria RAM
- 8 slot per espansioni ed interfacce
- alimentatore già dimensionato per la massima espansione.

In uno degli slot può essere inserita la scheda contenente il Language System Pascal, che amplia la memoria RAM a 64 K bytes e permette l'uso del linguaggio Pascal.
La macchina base comprende anche una uscita in video frequenza per il collegamento ad un monitor (o ad un comune televisore, a colori o in bianco e nero, previa installazione di un modulatore), ed un dispositivo analogico per il collegamento ad un registratore a cassette.
È l'unico personal computer ad essere dotato di vere possibilità grafiche, simili ad un plotter; infatti il video dell'Apple ha due possibilità grafiche a colori:

- a bassa risoluzione, con una matrice di 1880 punti a 16 colori
- ad alta risoluzione, con una matrice di 54000 punti a 6 colori.

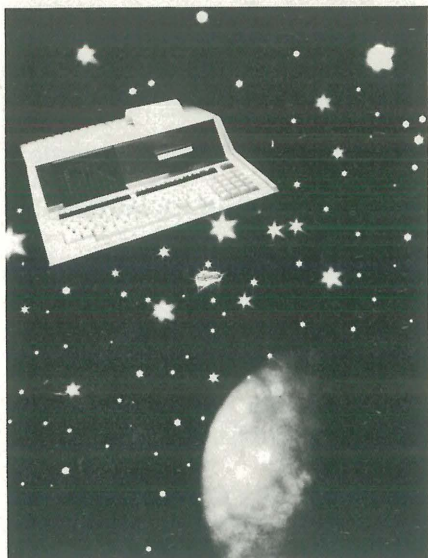
Per mezzo di interfacce che trovano posto negli slot della piastra, è possibile il collegamento con numerose periferiche:

- mini floppy disk della capacità di 116 o 143 K bytes
- stampanti con interfaccia parallela o seriale tipo RS232C
- scheda di comunicazione per invio dati all'esterno
- tavoletta grafica per introduzione disegni
- modem o accoppiatore acustico per trasmissione dati su linea telefonica
- convertitore analogico-digitale a 16 canali
- scheda orologio calendario al quarzo
- scheda PAL per la generazione del colore
- sintetizzatore musicale
- sintetizzatore vocale
- scheda speechlab, per riconoscimento vocale
- input analogici con risoluzione a 8 bit

Il personal computer Apple II è corredato da un'ampia dotazione di manuali che ne rendono l'uso e la programmazione molto semplice.
La grandissima diffusione dell'Apple II Plus nel mondo ha favorito sia la produzione di numerose interfacce compatibili Apple da parte di varie società, sia lo sviluppo di numerosi programmi applicativi di tipo professionale, tecnico, gestionale ed hobbystico, molti dei quali sono immediatamente utilizzabili anche sul mercato italiano.
È disponibile un catalogo esauriente in italiano, che verrà inviato a chi ne faccia richiesta.

IRET s.r.l. - REGGIO EMILIA

La Hewlett-Packard presenta il personal computer per i professionisti

Un nuovo sistema di calcolo personale con capacità grafiche incorporate, a basso costo, è stato introdotto in questi giorni dalla Hewlett-Packard.
Il nuovo sistema di calcolo, denominato HP-85, riunisce in una unità completamente integrata la potenza del suo processore centrale, la tastiera simile a quella di una normale macchina per scrivere, il video, la stampante, l'unità a cartuccia e le notevoli capacità grafiche.
Per utilizzare questo nuovo personal computer non è necessario avere un'esperienza precedente nell'uso di strumenti informatici in quanto la programmazione in linguaggio BASIC ne rende l'uso estremamente facile.
La parte numerica della tastiera, composta da 20 tasti, semplifica notevolmente l'introduzione dei

dati e l'esecuzione delle operazioni aritmetiche.
L'HP-85, prodotto dalla Corvallis Division della Hewlett-Packard, è stato progettato in funzione delle esigenze di calcolo personale di tecnici e progettisti, scienziati, commercialisti e altri professionisti del settore sia commerciale che industriale.
Inoltre, grazie al prezzo contenuto, alla facilità di impiego e alla compattezza, può essere utilizzato dagli hobbisti e nell'insegnamento nelle scuole medie superiori e nell'università.
Grazie alle quattro porte di input/output che consentono di collegare una vasta gamma di moduli di interfaccia opzionali, l'HP-85 si caratterizza non solo per le capacità di calcolo e grafiche all'avanguardia, ma anche per le potenti prestazioni nelle applicazioni di acquisizione e controllo dati.
Inoltre, le porte di input/output consentono all'utente di applicare il sistema aggiungendovi plotter, stampanti, unità a disco e le altre periferiche che diverranno man mano disponibili.

Memoria

La configurazione di base dell'HP-85 ha 16K byte di memoria di lettura/scrittura, di cui 14,5 K a disposizione dell'utente. Questa capacità di memoria può essere portata a 32 K byte, 30.5K dei quali a disposizione dell'utente, semplicemente innestando un modulo di memoria opzionale in una delle porte di input/output del calcolatore.

Linguaggio BASIC

L'HP-85 è programmabile in BASIC, un linguaggio interpretativo che risponde e va oltre alle più recenti prescrizioni ANSII, mettendo a disposizione dell'utente caratteristiche quali una precisione di 12 cifre, capacità di eseguire operazioni su stringhe e di editing, 42 funzioni predefinite, quattro livelli di sicurezza di programma e una formattazione dell'uscita molto flessibile.

Capacità grafiche

Un'importante caratteristica incorporata nell'HP-85 è la capacità di lavoro grafico in modo interattivo. Per rendere più chiare e comprensibili le informazioni, infatti, l'utente può rappresentare sullo schermo i dati in forma grafica.
Ad esempio, un tecnico può verificare i risultati di una serie di prove e di calcoli tracciando una curva ed analizzandone la distribuzione sullo schermo; un utente commerciale può esaminare l'andamento e le tendenze nelle sue operazioni semplicemente dando un'occhiata ad un disegno o ad una curva invece di esaminare lunghi elenchi di cifre.
Inoltre, tutto ciò che è rappresentato sullo schermo può essere riprodotto dalla stampante incorporata, un'operazione che può esere eseguita semplicemente premendo un tasto.

Tastiera

In funzione della comodità dell'utente, la tastiera è stata suddivisa in quattro gruppi di funzioni logiche: sezione alfanumerica simile alla tastiera di una macchina per scrivere, per l'introduzione dei dati numerici e l'esecuzione delle operazioni aritmetiche di somma, sottrazione, moltiplicazione, elevazione a potenza e divisione; tasti software cui l'utente può assegnare specifiche funzioni durante lo sviluppo di programma; tasti di controllo del sistema, delle funzioni di editing e del video che permettono all'utente di controllare lo schermo, il sistema operativo, l'unità a cartuccia e la stampante.

Video

Quando lavora nel modo alfanumerico, il video da 5 pollici, ad elevata risoluzione, può visualizzare fino a 16 linee di dati per volta, ciascuna composta da fino ad un massimo di 32 caratteri.
Ma la sua memoria è capace di contenere fino a 64 linee di dati, ciascuna delle quali può essere visualizzata sullo schermo facendo scorrere l'immagine verso l'alto o verso il basso.
Quando invece lavora nel modo grafico, il video diventa una matrice a punti, con un campo di 256 × 192 punti.
Ciò significa che l'utente dispone di 49152 punti distinti per i propri grafici ad alta risoluzione.
Inoltre, l'HP-85 memorizza l'ultima immagine sia alfanumerica che grafica, consentendo all'utente di passare da un modo di funzionamento all'altro senza che i dati vadano perduti nel passaggio.

Stampante

La stampante termica, alfanumerica e grafica, stampa silenziosamente alla velocità di 2 linee di 32 caratteri al secondo. Quando funziona nel modo alfanumerico, può stampare l'intero set di 128 caratteri ASCII, comprendente lettere maiuscole e minuscole, numeri e simboli speciali.
Ciascun carattere del set può inoltre essere sottolineato, consentendo all'utente di disporre di un set di 256 caratteri diversi.
Quando invece funziona nel modo grafico, la stampante può riprodurre qualunque rappresentazione grafica presente sullo schermo, sia sotto il controllo del programma che su comando da tastiera.
Se lavora nel modo grafico, la stampante fa compiere all'immagine presente sullo schermo una rotazione ideale di 90 gradi consentendo quindi la stampa di grafici continui, praticamente senza interruzione.

Unità nastro a cartuccia

L'unità nastro a cartuccia dell'HP-85 fornisce all'utente un comodo mezzo per la memorizzazione di dati e programmi che possono essere facilmente reperiti.
L'unità utilizza le cartucce per dati HP, con una capacità di 217K byte a una velocità di lettura/scrittura di 10 pollici al secondo e una velocità di ricerca di 60 pollici al secondo.
L'HP-85 provvede in modo automatico ad impostare all'inizio di ciascun nastro un indice dei contenuti e può quindi, in seguito, trovare automaticamente l'esatta locaziione dei dati e dei programmi memorizzati sul nastro stesso.

Software applicativo

Sono nove i packages applicativi già disponibili su cartucce preregistrate, mentre altri sono in fase di sviluppo; comunque è anche possibile adattare all'impiego sull'HP-85 i programmi BASIC sviluppati su altri computers, da tavolo della Hewlett-Packard.
Inoltre, poichè il linguaggio BASIC dell'HP risponde alle specifi-

che ANSII, è possibile adattare all'impiego sull'HP-85 anche la maggior parte del software compatibile con quegli standard.
Il software applicativo, oggi disponibile su cartucce preregistrate, comprende statistica generale, matematica ingegneria elettrotecnica, finanza, programmazione lineare e analisi di regressione.
Portatile
L'HP-85 misura 56 × 48 × 23 cm. e pesa 8 Kg.: per questo è adatto alle applicazioni che richiedono l'impiego di strumenti realmente portatili.
Insieme alla macchina, viene consegnato un manuale di 350 pagine che descrive il suo funzionamento e la programmazione, insieme ad un software per applicazioni standard che comprende 15 programmi.

HEWLETT-PACKARD - CERNUSCO S.N.

Microcalcolatore Ohio Scientific

Il C3-OEM è un microcalcolatore della Ohio Scientific che comprende in un unico contenitore alimentatore, cestello ad 8 posti scheda e due floppy disk drives. Caratteristica della macchina è l'utilizzazione di 3 microprocessori: il 6502, il 6800 e lo Z80, con la possibilità di deviare l'interpretazione delle istruzioni in memoria da un microprocessore ad un altro su comando software. Ne derivano due vantaggi: quello di poter sviluppare programmi nuovi facendo uso di volta in volta del microprocessore che si ritiene più adatto alla propria applicazione e quello di poter utilizzare su questa macchina tutto il software eventualmente già sviluppato o comunque disponibile da fonti diverse.
I floppy disk usati permettono di avere in linea 500 Kbytes, mentre la memoria ram, espansibile in blocchi di 16 K, è fornita delle dimensioni di 32 K statica.
L'entrata/uscita seriale RS 232 opera a velocità che vanno dai 75 a 19000 baud ed è disponibile inoltre una entrata/uscita a 16 linee.
Il software fornito comprende su ROM il bootstrap per il dischetto e i monitors di linguaggio macchina per il 6502 e il 6800 e su disco il BASIC e l'assembler/editor interattivo.
Tenendo presente che 4 posti scheda sono lasciati liberi, possono essere forniti i seguenti accessori:
- scheda da 16 K RAM statica o dinamica
- scheda di I/O con A/D, D/A, entrata/uscita parallela e seriale
- scheda di entrata/uscita vocale (votrax)
- scheda di entrata/uscita parallela a 96 linee
- scheda di I/O con 2 a 16 porte seriali
- interfaccia per unità a disco Winchesterda 23 a 74 megabytes.

EDICONSULT - MONZA

SOFTWARE

Al CLZ80 della SGS-ATES si aggiunge il BASIC

L'ultima aggiunta al software del CLZ80 realizzato dalla SGS-ATES è il BAS-Z: un interprete BASIC da 8K.
Si tratta di una versione del BASIC compatibile con il microprocessore Z80, sviluppata espressamente per il sistema microcalcolatore CLZ80.
Le sue caratteristiche sono paragonabili a quelle del BASIC di un minicomputer o di macchine ancora più complesse.
Il BAS-Z è dotato di un numero di funzioni che ne ottimizzano l'impiego per controllo di processo industriale, training su linguaggi ad alto livello ed usi amatoriali.
Questo BASIC da 8K byte è residente su 4 EPROM 2716 e può essere utilizzato anche con la configurazione di memoria minima del sistema microcalcolatore CLZ80.

SGS-ATES - AGRATE BRIANZA

Tre nuovi prodotti Microsoft

Una nuova organizzazione, la Vector Microsoft, è stata creata unitamente dalla Vector International (Belgio) e dal Microsoft (Bellevue, Washington) per sviluppare i servizi di vendita e di mantenimento dei prodotti Microsoft in Europa.

La Microsoft, pioniere mondiale nei sistemi software per microprocessori, ha ultimamente messo a punto tre nuovi prodotti BASIC. Il primo è la versione 5.0, la quinta generazione dell'interprete BASIC-80, compatibile con le norme ANSI; il secondo è un nuovo compiltore BASIC compatibile con la versione 5.0. Si tratta di un generatore di codici molto efficiente, più veloce da 5 a 10 volte del relativo interprete. Insieme al compilatore sono forniti il MACRO-80, un macro assemblatore Microsoft, e il caricatore d'agganciamento L80.
Il terzo prodotto nella nuova gamma del Vector Microsoft il BASIC-86, ugualmente compatibile con il BASIC-80 ed il primo interprete per i nuovi microprocessori a 16 bits. Il BASIC-86 è ora disponibile come disco indipendete concepito per Intel SBC 86/12. Questo prodotto assicura il trattamento d'archivi e tutti le funzioni d'utilità su disco.
Tutti i prodotti Microsoft, comprendenti anche i compilatori FORTRAN-80 e COBOL-80, sono dispobili su "floppy-disk" per i sistemi operativi CP/M e ISIS/II. La loro concezione modulare ne rende particolarmente facile l'installazione sui sistemi di lavoro OEM.
La Vector Microsoft offre una biblioteca di programmi standard e assicurerà il servizio delle installazioni OEM dalla sua sede in Belgio.
L'indirizzo della Vector Internazional è: Research Park, B-3044 Haasrode, Belgio (Tel.: +32 (16) 230894; Telex: 61978 Jewel b).

PACKAGE APPLICATIVI

Più estesa la biblioteca applicativa GP

La General Processor a reso disponibile, direttamente o tramite aziende collegate, una vasta biblio-

teca software per applicazioni di carattere commerciale. Tra i vari packages meritano un posto di preminenza quelli relativi alla gestione del magazzino e quelli di contabilità generale, che sono stati studiati per specifiche applicazioni e che quindi corrispondono esattamente alle più comuni esigenze dell'utenza commerciale ed amministrativa.
In virtù di questi nuovi pacchi applicativi l'impiego del microcomputer si è assai allargato anche a quella parte di utilizzatori che, per vari motivi, desiderano passare quanto prima, senza dovere subito procedere direttamente alla stesura dei programmi, a concrete e pratice applicazioni.

GENERAL PROCESSOR - FIRENZE

Nuovi programmi disponibili per l'Apple II

Sono ora disponibili per gli utilizzatori Apple due nuovi prodotti software di basso costo e di interessante utilità.

- Applepost: un package per gestione di un archivio anagrafico e per la stampa di indirizzi su buste e su etichette.
  Un programma molto facile da usare, ma notevolmente sofisticato, strutturato in modo da trarre il massimo vantaggio dalle notevoli possibilità dell'Apple II con mini floppy disk.
  Con l'Applepost è possibile introdurre e ritrovare liste di nomi ed indirizzi, stampare liste di etichette postali, e generare elenchi telefonici.
  Vi sono speciali routines di ricerca che permettono di ritrovare nomi che erano stati introdotti in modo errato, e di trovare indirizzi che siano in vicinanza geografica con altri.
  Interessante la possibilità di cercare tutti i nomi che abbiano assonanza con il nome richiesto.
  È stato scritto in modo da poter aumentare lo spazio di memorizzazione al crescere delle necessità, potendo arrivare a gestire fino a 6 driver di dischi contemporaneamente.
  È possibile registrare circa 500 nomi ed indirizzi su ogni disco.
- Visicalc: è un sistema di calcolo di uso generalizzato.
  Pianificare, progettare, calcolare, registrare, cancellare, classificare è possibile con il Visicalc in una frazione del tempo e dello sforzo necessario altrimenti.
  Infatti permette di risolvere tutti quei problemi che vengono normalmente elaborati con carta, penna e calcolatrice da tavolo.
  Visicalc combina la convenienza e familiarità di una calcolatrice con la potenza della memoria e le possibilità di visualizzazione del personal computer.

Il video dell'Apple diventa una "finestra" su un grande "foglio di carta" elettronico, suddiviso in righe e colonne.
Inoltre il computer ricorda dati, costanti e formule scritte di volta in volta, memorizzandole su diskette.

SOFTEC - MILANO

CORSI

La Softec, nell'ambito della propria attività di supporto sistemistico al personal computer Apple II, ha organizzato seminari di presentazione e corsi specifici di BASIC-,DOS, e di Pascal.
I seminari, della durata di mezza giornata e completamente gratuiti, vengono tenuti ogni settimana.
Gli argomenti:
- cosa è un personal computer
- possibili configurazioni del sistema
- caratteristiche tecniche ed operative
- introduzione al linguaggio BASIC
- il Sistema Operativo per la gestione di informazioni su disketti (DOS)
- presentazione del linguaggio Pascal
- esempi di applicazioni
- consulenza tecnica

Programma del corso BASIC, della durata di un giorno:
- la struttura del linguaggio BASIC
- comandi di sistema e di utilità
- comandi di editing e di formattamento
- trattamento stringhe e matrici multidimensionali
- l'input/output
- le frasi di controllo
- gestione della grafica e dispositivi accessori
- funzioni matematiche
- controllo e gestione degli errori
- utilizzazioni del BASIC con esempi applicativi
- ottimizzazioni del programma BASIC

Programma del corso DOS, della durata di un giorno:
- struttura del sistema operativo
- formato dalle informazioni su disco
- l'allocazione dinamica dei files
- comandi di uso generale
- sicurezza dell'informazione
- files sequenziali
- files ad accesso casuale
- procedure catalogate per esecuzione automatica di sequenze di programmi
- files binari
- concatenamento di programmi
- gestioni di comandi DOS da programma BASIC

Programma del corso Pascal, della durata di 3 giorni:
- il linguaggio Pascal
- concetto di macchina Pascal
- struttura del programma
- il concetto di dato
- le dichiarazioni
- il concetto di azione
- le frasi esecutive
- altri tipi di dati (record, set, array, ...)
- procedure e funzioni
- input/output
- il concetto di librerie di sistema (frase Uses)
- strumenti di sviluppo
- files management
- editor
- compilatore
- link
- debug
- esecuzione di un programma
- il Pascal su personal computer
- il macro-assembler rilocabile e linkabile con moduli Pascal.

La Softec è disponibile per ulteriori informazioni e prenotazioni presso le proprie sedi di:

Milano tel. 02/3490231
Torino tel. 011/6509303

Il Microprocessore in aiuto agli handicappati

In collaborazione con la rivista Elettronica Oggi, è stato indetto il concorso "Il microprocessore in aiuto ai portatori di handicap".
Il recente sviluppo delle tecnologie elettroniche ha permesso la progettazione e la realizzazione di dispositivi in grado di fornire un supporto tecnico ai portatori di handicap.
Al fine di promuovere e raccogliere studi, idee a realizzazioni sviluppate per mettere il portatore di

handicap in grado di superare alcuni limiti causati dalla sua invalidità, è stato indetto un concorso aperto a tutti i progettisti elettronici.

Esso verte sulla presentazione di progetti e applicazioni utili quali ausili per disabili (motulesi, non vedenti, audiolesi e persone con difficoltà di espressione e/o movimento) sia nella fase diagnostica (valutazione della gravità dell'handicap), che riabilitativa.
Saranno prese in considerazione anche altre applicazioni non convenzionali dei microprocessori non strettamente rivolte all'oggetto del concorso purchè di sicuro interesse e sempre in campo elettromedicale. Esse concorreranno all'assegnazione di premi speciali.
I progetti dovranno essere presentati con: schema a blocchi e circuitale dell'applicazione, progetto hardware, software, costi, dati fisici (dimensioni, pesi ecc.).
Essere corredati con una descrizione tecnico scientifica atta a presentare i vantaggi e la funzionalità dell'applicazione.
La presentazione di un eventuale prototipo dell'applicazione è auspicabile anche se non indispensabile.
Gli elaborati dovranno pervenire alla Segreteria del concorso entro il 20 maggio 1980.
I progetti saranno analizzati e valutati da una commissione tecnico scientifica di estrazione interdisciplinare.
I vincitori saranno premiati ed i loro lavori presentati domenica 8 giugno 1980 nel corso della mostra BIAS '80 Microelettronica, che si terrà dal 4 all'8 giugno 1980 alla Fiera di Milano.
L'organizzazione del concorso si farà parte dirigente per promuovere gli eventuali ulteriori sviluppi delle idee e delle proposte più significative attraverso collegamenti con Università ed Enti di ricerca pubblica e privata, nazionale ed internazionali, salvaguardando beninteso la proprietà del brevetto al progettista.
Il concorso ha una dotazione globale di 7.000 dollari, oltre a premi costituiti da sistemi, strumenti ed apparecchiature elettroniche avanzate.
Per ulteriori informazioni sul concorso, si prega di telefonare o scrivere a:
Studio Barbieri, Viale Premuda, 2 - 20129 MILANO - Tel. 796.096/421/635 (02).

Siamo stati i primi . . .

a proporre un oscilloscopio professionale sotto il "Tetto" delle 500.000 lire. Ricordate il vecchio 12-4DA? è ancora il nostro più accanito concorrente: infatti chi l'ha acquistato (e sono stati in molti) non vuol saperne di cambiarlo. Ma guardate:

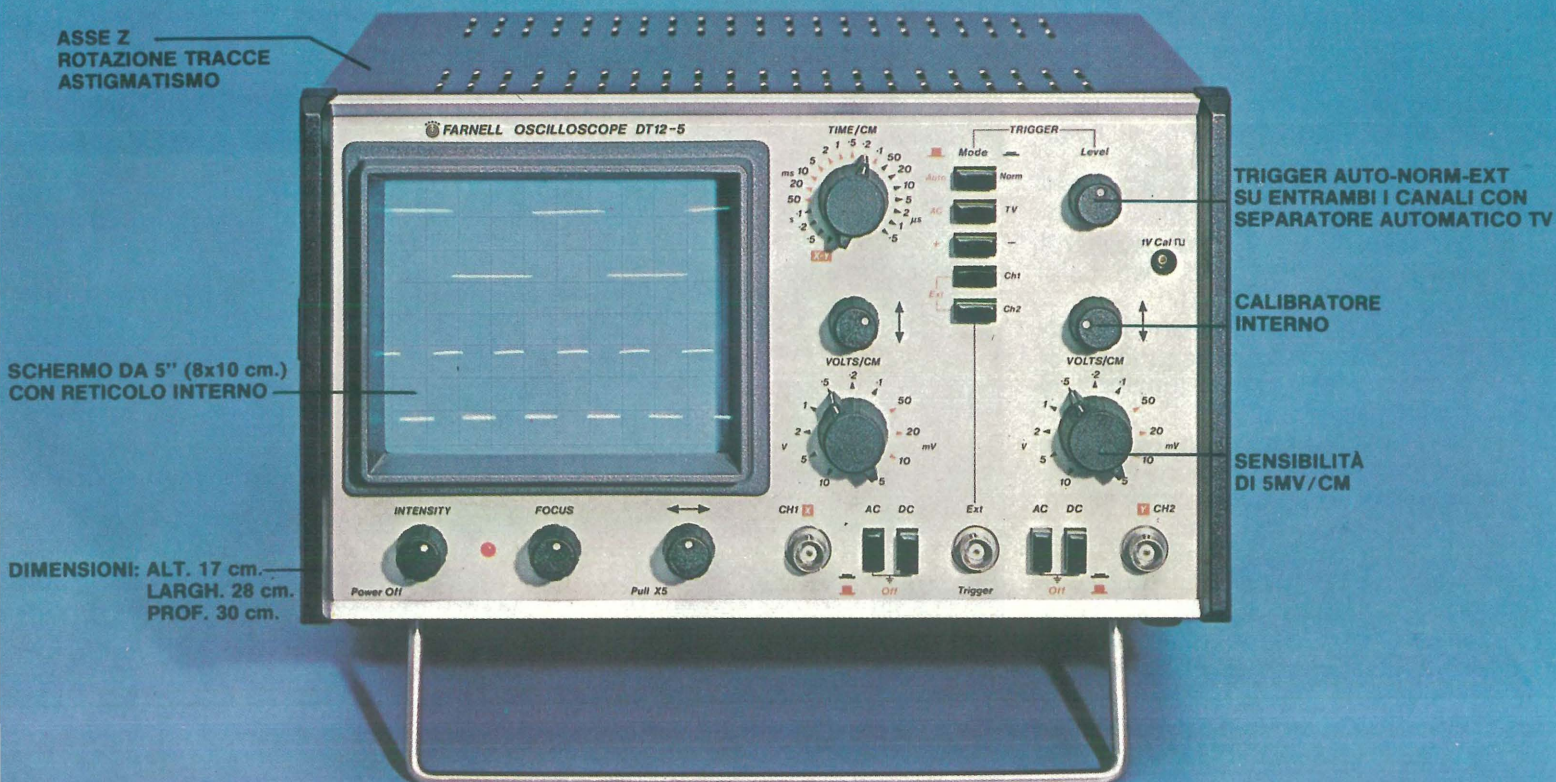
ASSE Z
ROTAZIONE TRACCE
ASTIGMATISMO

TRIGGER AUTO-NORM-EXT
SU ENTRAMBI I CANALI CON
SEPARATORE AUTOMATICO TV

CALIBRATORE
INTERNO

SCHERMO DA 5" (8x10 cm.)
CON RETICOLO INTERNO

SENSIBILITÀ
DI 5MV/CM

DIMENSIONI: ALT. 17 cm.
LARGH. 28 cm.
PROF. 30 cm.

. . Ora siamo gli unici

in grado di offrirvi una nuova generazione di oscilloscopi europei a doppia traccia, 12 MHz, ultracompatti (grazie al nuovo, ridottissimo, CRT che la Brimar ha sviluppato per noi) al prezzo di

486.000 LIRE

Farnell Italia s.r.l.

Via Mameli, 31 - 20129 MILANO - Tel. (02) 7380645 - 733178

DISPONIBILE A STOCK PRESSO:

SASSUOLO	- HELLIS	Tel. 059/804104
TORINO	- CARTER	Tel. 011/592512
CHIAVARI	- GOLD	Tel. 0185/300773
ROMA	- SILV	Tel. 06/8313092
NAPOLI	- E.D.L.	Tel. 081/632335
BOLZANO	- RADIOMARKET	Tel. 0471/37407
TRIESTE	- RADIOKALIKA	Tel. 040/30341
CATANIA	- THYRISTOR	Tel. 095/444581

Viste le caratteristiche, pensateci un po' non ne vale la pena?

☐ Desidero avere maggiori informazioni
☐ Desidero riservare il mio DT 12-5. Vogliatemi confermare le condizioni di acquisto.

Nome . . . .
Cognome . . . .
Ditta . . . .
Via . . . . N° . . . .
Città . . . . CAP . . . .
Tel. . . . .

*Prezzi validi al 31/12/79 IVA Esclusa Pag. alla consegna.

FEEDBACK

Se avete pensato che avessimo pubblicato soltanto le lettere e le critiche che più ci facevano comodo, ebbene, vergognatevi. In realtà, lo ammetto, l'idea era allettante, ma anche noi abbiamo un qualche brandello di coscienza, abbastanza da riconoscere con noi stessi e con voi che seppellire una critica non significa rimuovere il difetto che ne sta alla base. Così, anche se ci siamo un pò montati la testa per via dei complimenti che avete avuto l'imprudenza di farci, tentiamo almeno di considerare con uguale obiettività anche la le critiche che avete avuto la crudeltà di muoverci.

Naturalmente scherzo: ascoltare una critica è un pò come alzarsi da letto la mattina, spiacevole per un breve attimo, ma condizione indispensabile per fare qualcosa di costruttivo. E veniamo a voi.

Fabio Fumi

Alcuni ... suggerimenti

Accolgo il Vostro invito a esprimere giudizi e a dare suggerimenti.

I numeri finora apparsi mi sono sembrati interessanti per il contenuto in senso generale, anche se si può notare che non esiste ancora una precisa configurazione dei contenuti.

Diversi articoli sono di autori stranieri e ciò lo ritengo valido vista la più lunga esperienza nel settore.

La pubblicità sulla rivista va molto bene, purchè sia sempre ben distinta dal contenuto redazionale.

Dal punto di vista grafico nulla da eccepire.

Qui di seguito Vi elenco una serie di suggerimenti:

1) Stabilire a chi debba rivolgersi la rivista (progettisti, utenti, hobbysti, ecc.). Ovviamente è possibile accontentare diversi tipi di lettori, però con rubriche distinte. Personalmente sono un hobbysta a livello di uso dei microcomputers.
2) La rivista dovrebbe presentare con una serie di articoli tutti i linguaggi di programmazione, evidenziandone differenze, caratteristiche, pregi e difetti.
3) Dei linguaggi di programmazione più in uso (o interessanti) dovrebbero essere sviluppati dei veri e propri corsi con andamento ciclico e di difficoltà crescente.
4) Argomento che dovrebbe essere sviluppato è quello relativo alle tecniche di analisi e programmazione (insiemistica, programmazione strutturata, ecc.).
5) Indispensabili articoli in cui vengano illustrati i microcomputers presenti sul mercato italiano. Tali articoli dovrebbero essere molto critici e di conseguenza essere tratti da prove effettive fatte in redazione da esperti.
   È importante che questi articoli mettano in evidenza le reali caratteristiche tecniche (hardware), le possibilità di utilizzo di periferiche, il software, disponibile, il rapporto qualità/costo, il rapporto memoria/costo, l'assistenza, ecc.
6) Lo stesso discorso di cui al punto precedente, però riguardante le periferiche (floppy disks, stampanti, registratori, ecc.).
7) Rubrica dei lettori in cui possano essere posti quesiti tecnici di scelta, oppure dove possano essere evidenziati reclami o denunce su particolari argomenti (inconvenienti, assistenza, ecc.).
8) Rubrica di recensione libraria.
9) Articoli in cui vengano presentati programmi (col listing) di giochi e di appliazioni di interesse generale.
10) Articoli che si rivolgano all'utente hobbystico per consigliarlo sulla scelta delle apparecchiature. Ovviamente non suggerendo marche, ma caratteristiche confacenti all'uso che ne dovrà fare.
11) Rubrica dove i lettori presentano loro programmi (col listing).
    Tali programmi dovrebbero essere scelti dalla redazione dopo verifica.
    Volendo potrebbero essere premiati quelli pubblicati, con un abbonamento, o ...
12) Rubrica di presentazione di tutte le novità tecniche uscite, possibilmente con fotografie.

L'elenco è piuttosto lungo e forse poteva continuare, ma mi fermo per darVi un solo ulteriore importante suggerimento: la rivista è bene che esca con periodicità mensile.

Cordiali saluti.

Rodolfo Spinosa, Segrate

Hai molta ragione su diversi punti.

È vero che non esiste ancora una precisa incasellatura dei contenuti, anche perchè è piuttosto difficile, nel senso che non sempre è ovvio in quale sezione vada messo un determinato articolo, temo che la voce "varie" ne risulterebbe stracolma a spese delle altre. Ciò non significa che non si debba provare, avendo l'avvertenza di organizzare le cose in maniera che ognuno legga anche un pò fuori dal suo campo specifico: ho parlato troppo spesso con degli specialisti all'americana, ferratissimi nel loro campo ed escusivamente in quello, perchè la cosa mi piaccia; il discorso interdisciplinare rischia di risultarne soffocato, col risultato di produrre una serie di "raggionatori a compartimenti stagni" con una brutta riduzione della flessibilità mentale. Abbi pazienza per un pò, organizzare una cosa simile è un lavoro che va fatto bene, con una certa calma.

Ed ora, i punti numerati:

1) *BIT si rivolge a chiunque abbia un minimo di preparazione specifica nel campo dei computers ed un robusto interesse nello stesso campo, dallo studente liceale colpito da informatica epidemica in forma grave, in su. Si rivolge prevalentemente ai giovani (mentalmente parlando, ossia a coloro che pensano che al mondo ci sia sempre molto da imparare) mentre non è adatto a coloro che pensano di sapere già tutto quello che occorre, per obiettive difficoltà tecniche nel far leggere qualcosa a costoro. Capirai che accontentare tanti lettori così disparati, senza scoraggiare lo studente o annoiare il professore universitario, non è semplice; ma vogliamo riuscirci.*
2) *Lo faremo. Anche se ho qualche perplessità sul "tutti", se in questi dobbiamo includere anche i diversi assembler dei disparatissimi micro; forse arriveremo onorevolmente ad un "quasi tutti".*
3) *Chi stabilirà quali sono i linguaggi più interessanti? Per il Pascal stiamo già facendo qualcosa di simile, ma per il resto dovrete chiederceli voi. Chiedete e vi sarà dato.*
4) *Sono argomenti abbastanza indeterminati ogni analista ha da dire la sua in proposito ed in generale non è d'accordo con gli altri. Lo faremo se riusciremo a mettere d'accordo più di tre analisti senza usare la violenza.*
5) *OK, verissimo. Sia per i micro propriamente detti, che per il supporto e la preparazione e l'accessibilità delle ditte che li trattano. Magari ci faremo dei nemici, ma credo anch'io che sia indispensabile fare questo discorso.*

6) *Più complicato del punto precedente, perché le possibili periferiche sono molte di più e gli usi diversi di una stessa periferica la possono inquadrare come buona, mediocre o scadente a seconda che le sue criticità influiscano o meno sull'uso che se ne fa. L'impegno economico relativo è molto maggiore di quello richiesto da un lavoro analogo fatto sui micro. Tutto dipende quindi da Chi tiene i cordoni della borsa, proveremo a parlarGliene con la dovuta delicatezza.*
7) *La stai leggendo proprio adesso. Se un argomento fosse troppo grosso per starci, varrebbe la pena di farne un articolo a sé stante, anche se breve. Non abbiamo alcuna prevenzione contro gli articoli scritti dai lettori, tutt'al più una recondita paura che siano migliori dei nostri.*
8) *Non fare richieste imbarazzanti: la Jackson è un'editrice anche di libri tecnici, l'obiettività quindi verrebbe sempre messa in dubbio, persino se ci sforzassimo di essere cattivi con la Jackson. Ma non è detto che non si faccia.*
9) *Ci sono già.*
10) *Forse è meglio aspettare ancora un pò: le apparecchiature a portata del portafogli di un hobbista sono ancora molto poche, come tristemente ben sai.*
11) *Perchè no?*
12) *Faremo del nostro meglio: ma, come novità, intendi i prodotti di cui ti fanno vedere un esemplare gelosamente conservato in una teca di cristallo, o un prodotto che si possa anche acquistare? C'è una bella differenza, talvolta anni.*
13) *Stiamo appena proponendo di ampliare lo staff; per un pò ancora dovrai rassegnarti a leggerci ogni due mesi.*

Grazie delle propote; le buone idee sono il pane di una rivista.

A me va bene così!

Accolgo il vostro invito dell'editoriale sul n. 4.

Ho particolarmente apprezzato la vostra rivista.

Per mio conto la rivista è ben impostata divisa per argomenti così com'è. Chi dei micro ne fa un hobby può, a mio parere, essere di due categorie: il personal computer può essere visto come "mezzo" dell'hobby oppure come "fine". Alla prima categoria, per esemplificare, potremmo ascrivere chi ha un PET o simili, insomma uno "scatolone" senza contatti HW verso l'esterno, un softwarista (se permettete l'espressione) puro insomma. Alla seconda chi intorno a un single-board o simili si costruisce il "proprio" computer.

Appartenendo (in potenza: non ho un micro! ... ancora) alla seconda categoria, è logico che ritenga troppo lo spazio dedicato ai programmi-gioco.

In particolare potrei rimproverare il fatto che gli articoli non sono orientati ad insegnare finezze (e non) di programmazione, ma semplicemente piatte presentazioni di un gioco e, mi si scusi se sbaglio, credo che pochi "utenti" si siano sforzati di capir bene il programma (oltrettutto penso che un programma mal commentato, o peggio non commentato affatto, sia utile solo al suo programmatore).

Ho apprezzato particolarmente gli articoli sul Pascal e sulla programmazione strutturata, e spero che non siano gli ultimi sull'importante argomento.

Ancora ... mi interessa particolarmente la presentazione delle schede, ma se son fatte dall'importatore/rivenditore hanno tutti i pregi in bella evidenza e i difetti nascosti tra le righe ... oltre a riportare raramente una pur vaga indicazione sul prezzo.

Dulcis in fundo ... è obbiettivamente difficile per una rivista come BIT trovare il giusto compromesso tra semplicità e interesse di contenuti, caratteristica, potrei dire, della casa editrice. Però, avendo parlato con amici che come me hanno il virus dell'elettronica, ma solo per il tempo libero (io sono laureando in ingegneria e, tra l'altro, nella bistrattata università ho studiato sia il micro (8080, logico) che il Pascal) posso dirvi che hanno smesso di leggere BIT perchè troppo difficile. In compenso lo leggono molti miei compagni d'università (un 10% nella specialità di comunicazioni).

La stessa serie del picocomputer richiede espressamente le basi: insomma per leggere BIT è necessario avere poche ma chiare idee sui micro. Allora si potrebbe tollerare un articolo/numero sulle basi, ma c'è l'inconveniente che chi si avvicinerà a BIT in futuro ormai li avrà persi.

Non che ci siano semplici soluzioni al problema: voi fate del vostro meglio e nel mio piccolo non vi mancherà l'appoggio.

Pietro Molina, Vigevano

Noto una certa disparità di vedute fra te e Rodolfo Spinosa, cui rispondo nella lettera precedente; credo che il pluralismo sia qualcosa del genere ma, se non riusciamo a metterci d'accordo, andrà a finire che si dovrà fare una rivista diversa per ciascuno di voi. E pensare che all'inizio ci eravamo chiesti se ci fosse spazio abbastanza per BIT ...

È interessante la tua distinzione fra softwaristi puri e hardware-softwaristi (prego la Dea Grammatica che si scriva davvero così): una distinzione del genere, ai tempi dei grossi computers, vedeva i Sacerdoti che si occupavano del software diversi scalini più in alto dell'umile tecnico che metteva le mani fisicamente nelle budella del mostro e che non sapeva scrivere un rigo di programma. Dal momento che sono spaventosamente orgoglioso di essere stato uno dei primi uomini misti hardware/software, non posso che essere lieto di vedere che la mia gente si va espandendo, e triste per il fatto che ci sia anche nel campo dei micro chi sdegna di risolvere un problemino con un paio di gates esterni piuttosto che con un paio di dozzine di righe di programma. Comunque, sii un pò più tollerante: per il momento diamo ai pur softwaristi panem et circenses, *vedrai che un giorno riusciremo a tirarli dalla nostra e ad incuriosirli tanto nei confronti dei visceri del loro "scatolone" da farne dei veri maniaci come noi.*

Devo darti pienamente ragione circa l'utilità del commento in un programma, e non solo per quel che riguarda la comunicabilità fra programmatori diversi, quanto anche per la comunicabilità fra un programmatore e se stesso a qualche mese di distanza: ti è mai successo di scoprire di aver dimenticato che razza di diabolico ed astutissimo trucchetto avessi usato a suo tempo per risparmiare memoria? A me sì. Forse non tanto in un linguaggio ad alto livello, quanto in uno degli svariati assembler. E ti giuro che quando uno si accorge di non capire più se stesso i pensieri si fanno tristi e corrono ai danni della senescenza sul cervello umano.

La programmazione strutturata dovrebbe risolvere anche questi inconvenienti, ma ho forti paure che, se male usata, rischi di portare ad un appiattimento generale del software: una cosa è usare degli sporchi trucchi senza documentarli adeguatamente, un'altra è rinunciare ad essi del tutto per evitare i loro inconvenienti: rischia di diventare un pò come rinunciare del tutto al vino per evitare il mal di testa del giorno dopo la sbronza.

Il tuo dulcis in fondo gira il coltello nella classica piaga: una rivista come BIT presuppone un minimo di nozioni di base nel lettore; come spesso succede, a chi questo minimo lo conosce già, pare niente, mentre chi non ne ha la minima idea si lascia facilmente terrorizzare dalle tenebre dell'ignoto.

Nessuno dei due punti di vista corrisponde all'ipotetica realtà: se ci pensi un attimo, ti renderai facilmente conto che non c'è da conoscere dieci argomenti difficilissimi, per cominciare a leggere BIT, ma mille banalità. Il guaio è che anche per raccontare mille banalità ci vuole ben di più che una rubrica o un numero della rivista. Visto poi che molte di queste banalità si imparano più pasticciando al banco del laboratorio che non leggendole da un foglio stampato, il problema non ha soluzioni reali. Se però sono ammesse anche quelle compiesse coniugate, direi che potrebbe funzionare, per chi non abbia le basi, il metodo di farsi prestare BIT (per capirci poco non vale la pena di acquistarlo, che la Direzione mi perdoni) per un certo numero di volte, sorvolando su tutto quel che non è chiaro; contemporaneamente sforzandosi di realizzare qualcosa di concreto e funzionante con le proprie mani attingendo a BIT, ai data sheets delle diverse ditte e a tutte le altre fonti di informazione disponibili. Dopo un pò di terapia si finisce con l'acquisire l'inizio di una mentalità che permette di divinare la natura di un argomento di base, anche in assenza di spiegazioni esplicite, con una minima probabilità di errore.

Questo perchè ci sono sì molti modi di pelare un gatto, ma in elettronica, non so perchè, i migliori si assomigliano un pò tutti, quello che si riesce a ricostruire con la propria mente risulta spesso sorprendentemente simile a quello che altri, magari professionisti nel campo, hanno ideato separatamente.

Per parte nostra cureremo che non manchi in nessun numero una certa quantità di informazione sulle basi, abbastanza cospicua per interessare chi ne abbia bisogno, ma abbastanza piccola da non far sbadigliare che ne sa già abbastanza.

Ti ringrazio del tuo appoggio: piccolo o no, ci fa un grande piacere.

Interprete BASIC

Abbiamo ricevuto numerose lettere che ci hanno segnalato la mancanza di numerose righe nel listato pubblicato sul numero 5 di BIT. A dire il vero siamo stati anche minacciati o lusingati a seconda dei casi per stimolarci all'integrazione del listato, quasi avessimo avuto la deliberata volontà di ren-

dere inutilizzabile il listato stesso. A parte il fatto che questo non rientra nel nostro stile e se mai è vero il contrario, chiediamo venia. È stato al solito un errore fatto in fase tipografica.

L'interesse recepito tramite questo non voluto feedback ci permette comunque di segnalarvi il *nostro* interesse ad eventuali estensioni dell'interprete: è il minimo da chiedere a nome di tutti i lettori di BIT.

M.M.

```
853  ;
854  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
855  ;                                                                                  ;
856  ;                                                                                  ;
857  ;.........................EXPR.....................................................;
858  ;                                                                                  ;
859  ;                                                                                  ;
860  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
861  ;                                                                                  ;
862  ;                                                                                  ;
863  ; 'EXPR' VALUTA LE ESPRESSIONI ARITMETICHE E LOGICHE SECONDO QUESTA GRAMMATICA      ;
864  ; <EXPR> ::= <EXPR2> ! <EXPR2><REL.OP.><EXPR2>                                     ;
865  ; <REL.OP.> ::=  >=  <=  <  >  =  #                                                ;
866  ; IL RISULTATO DI UN'OPERAZIONE LOGICA E' 1 OPPURE 0 SE E' VERA O FALSA             ;
867  ; <EXPR2> ::= <EXPR22> ! +<EXPR22> ! -<EXPR22>                                     ;
868  ; <EXPR22> ::= <EXPR3> ! <EXPR22>+<EXPR3> ! <EXPR22>-<EXPR3>                       ;
869  ; <EXPR3> ::= <EXPR4> ! <EXPR3>*<EXPR4> ! <EXPR3>/<EXPR4>                          ;
870  ; <EXPR4> ::= <VAR> ! <FUNZIONE> ! (<EXPR>) ! <VETT> ! <NUM>                        ;
871  ; <VAR> ::= A B C D E F G H I J K L M N O P Q R S T U V W X Y Z                     ;
872  ; <FUNZIONE> ::= ABS(<EXPR>) ! RND(<EXPR>) ! SIZE                                   ;
873  ; <VETT> ::= @(<EXPR>)                                                             ;
874  ; <NUM> ::= <CIFRA> ! <NUM><CIFRA>                                                 ;
875  ; <CIFRA> ::= 1 2 3 4 5 6 7 8 9 0                                                  ;
876  ; IL SEGNO '!' SIGNIFICA 'OPPURE' . <EXPR> E' RECURSIVO                            ;
```

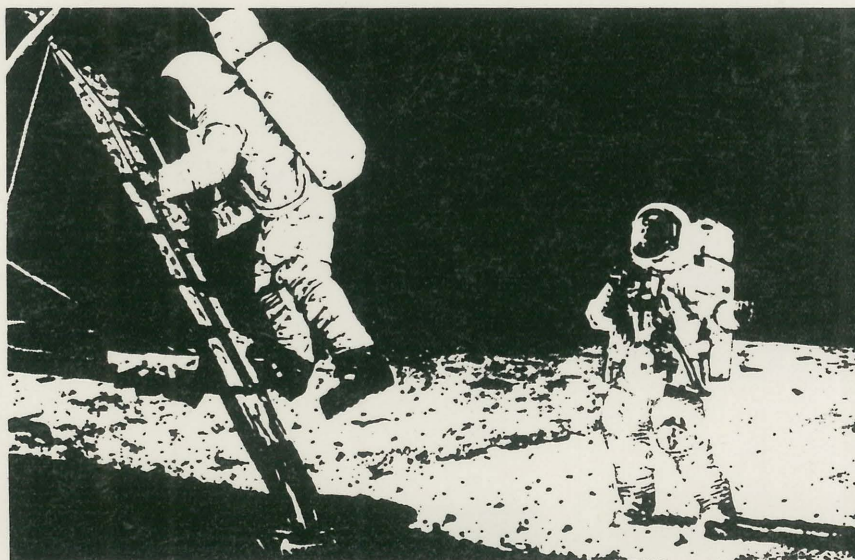
```
                 1146  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 1147  ;                                                                                  ;
                 1148  ;                                                                                  ;
                 1149  ;........SETVAL........FIN........ENDCHK........ERROR.(ET.ALII)...................;
                 1150  ;                                                                                  ;
                 1151  ;                                                                                  ;
                 1152  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 1153  ;                                                                                  ;
                 1154  ;                                                                                  ;
                 1155  ; 'SETVAL' SI ASPETTA UNA VARIABILE, UN SEGNO DI UGUALE ED UNA ESPRESSIONE :        ;
                 1156  ; VALUTA L'ESPRESSIONE E ASSEGNA IL VALORE TROVATO ALLA VARIABILE                  ;
                 1157  ;                                                                                  ;
                 1158  ; 'FIN' CONTROLLA LA FINE DI UN COMANDO. SE TERMINA CON ';' L'ESECUZIONE PRO-      ;
                 1159  ; SEGUE SULLA STESSA LINEA. SE IL COMANDO TERMINA CON UN CARRIAGE-RETURN,          ;
                 1160  ; CERCA LA LINEA SUCCESSIVA E PROSEGUE DI LI'.                                     ;
                 1161  ;                                                                                  ;
                 1162  ; 'ENDCHK' CONTROLLA SE UN COMANDO TERMINA CON UN CARRIAGE-RETURN. ALCUNI CO-       ;
                 1163  ; MANDI INFATTI DEVONO TERMINARE CON UN CARRIAGE-RETURN (ES. GOTO STOP ECC.)       ;
                 1164  ;                                                                                  ;
                 1165  ; 'ERROR' STAMPA LA STRINGA PUNTATA DA DE TERMINANTE CON UN CR-LF. STAMPA POI      ;
                 1166  ; LA LINEA PUNTATA DA 'CURRNT' CON UN '?' INSERITO ALLA POSIZIONE A CUI PUNTA-     ;
                 1167  ; VA IL TEXT POINTER QUANDO E' STATO RICONOSCIUTO L'ERRORE: IL TEXT POINTER        ;
                 1168  ; DEVE ESSERE SALVATO IN CIMA ALLO STACK QUANDO SI RICONOSCE L'ERRORE.             ;
                 1169  ; L'INTERPRETATORE  RIPARTE POI DA CAPO. SE PERO' CURRNT -> 0 ,SIGNIFICA CHE       ;
                 1170  ; SI E' SOTTO COMANDO DIRETTO (CIOE' NON SI E' IN ESECUZIONE DI PROGRAMMA) :      ;
                 1171  ; E IL COMANDO DIRETTO ERRATO NON VIENE STAMPATO. INFINE SE CURRNT -> A VALORI    ;
                 1172  ; NEGATIVI, SIGNIFICA CHE SI E' IN 'INPUT',LA LINEA SBAGLIATA NON VIENE RISTAM-   ;
                 1173  ; PATA, E IL CONTROLLO E' PRESO DA 'INPERR'                                        ;
                 1174  ;                                                                                  ;
                 1175  ; ALLA ROUTINE 'ERROR' SONO ASSOCIATI I SEGUENTI ENTRY POINT:                      ;
                 1176  ; 'QWHAT' SALVA IL TEXT POINTER E STAMPA 'WHAT?'                                   ;
                 1177  ; 'AWHAT' STAMPA SOLO 'WHAT?' SENZA SALVARE IL PUNTATORE                           ;
                 1178  ; ANALOGAMENTE  PER  'QSORRY' , 'ASORRY' , 'QHOW' , 'AHOW'.                         ;
                 1179  ; NOTARE CHE GLI ULTIMI DUE ENTRY POINT SONO IN UN'ALTRA SEZIONE DI PROGRAMMA      ;
                 1180  ;                                                                                  ;
                 1181  ;                                                                                  ;
05A6  CD9500   R 1182  SETVAL   CALL TSTV             ;CHIEDE 'WHAT?' SE NON E' UNA VARIABILE
05A9  DAD805   R 1183           JP C,QWHAT            ;SALVA L'INDIRIZZO DELLA VAR.
05AC  E5         1184           PUSH HL               ;SALVA L'INDIRIZZO DELLA VAR.
05AD  CDC900   R 1185           CALL TSTC             ;CONTROLLA LA PRESENZA DEL SEGNO =

                    TINY.BASIC      21$GIUGNO$1979      PAGE  22
 LOC  OBJ CODE M STMT SOURCE STATEMENT                        BAU 4.1

05B0  3D         1186           DEFB '='
05B1  0A         1187           DEFB (SV1-$-1)&0FFH
05B2  CD1D04   R 1188           CALL EXPR             ;VALUTA L'ESPRESSIONE
05B5  44         1189           LD B,H                ;METTE IL VALORE DELL'ESPRESSIONE IN BC
05B6  4D         1190           LD C,L
05B7  E1         1191           POP HL                ;RECUPERA L'INDIRIZZO DELLA VAR.
05B8  71         1192           LD (HL),C             ;ASSEGNA IL NUOVO VALORE ALLA VARIABILE
05B9  23         1193           INC HL
05BA  70         1194           LD (HL),B
05BB  C9         1195           RET
05BC  C3D805   R 1196  SV1      JP QWHAT              ;MANCA IL SEGNO '=' ;SEGNALA L'ERR.
                 1197  ;
05BF  CDC900   R 1198  FIN      CALL TSTC
05C2  3B         1199           DEFB ';'
05C3  04         1200           DEFB (FI1-$-1)&0FFH
05C4  F1         1201           POP AF                ;C'E' ';' TOGLIE L'IND. DI RIT.
05C5  C31F02   R 1202           JP RUNSML             ;CONTINUA AD ESEGUIRE LA STESSA LINEA
05C8  CDC900   R 1203  FI1      CALL TSTC             ;NON C'E' ';' :
05CB  0D         1204           DEFB CR               ;E' ALLORA UN CARRIAGE-RETURN ?
05CC  04         1205           DEFB (FI2-$-1)&0FFH
05CD  F1         1206           POP AF                ;SI',TOGLIE L'IND. DI RITORNO
05CE  C30F02   R 1207           JP RUNNXL             ;VA AD ESEGUIRE LA LINEA SUCCESSIVA
05D1  C9         1208  FI2      RET                   ;ALTRIMENTI TORNA AL CHIAMANTE
05D2  CD8600   R 1210  ENDCHK   CALL IGNBLK
05D5  FE0D       1211           CP CR                 ;TERMINA CON UN CARRIAGE-RETURN?
05D7  C8         1212           RET Z                 ;SI',RITORNA
                 1213                                 ;ALTRIMENTI STAMPA 'WHAT?'
05D8  D5         1214  QWHAT    PUSH DE
05D9  111701   R 1215  AWHAT    LD DE,WHAT
05DC  97         1216  ERROR    SUB A
05DD  CD8106   R 1217           CALL PRTSTG           ;STAMPA LA STRINGA PUNTATA DA DE
05E0  D1         1218           POP DE                ;CIOE' 'WHAT','HOW' O 'SORRY'
05E1  1A         1219           LD A,(DE)             ;SALVA IL CARATTERE A CUI ->
05E2  F5         1220           PUSH AF               ;IL PRECEDENTE DE ,
05E3  97         1221           SUB A                 ;E LO CAMBIA CON UNO ZERO
05E4  12         1222           LD (DE),A
05E5  2A0108   R 1223           LD HL,(CURRNT)        ;CONTROLLA IL NUMERO
05E8  E5         1224           PUSH HL               ;DELLA LINEA CORRENTE
05E9  7E         1225           LD A,(HL)
05EA  23         1226           INC HL
05EB  B6         1227           OR (HL)
05EC  D1         1228           POP DE
05ED  CA0000   R 1229           JP Z,START            ;SE E' ZERO RIPARTE SOLO DA CAPO
05F0  7E         1230           LD A,(HL)             ;SE E' NEGATIVO
05F1  B7         1231           OR A                  ;TORNA A LEGGERE UNA LINEA
05F2  FAA503   R 1232           JP M,INPERR           ;DALLA TASTIERA
05F5  CD0107   R 1233           CALL PRTLN            ;ALTRIMENTI STAMPA LA LINEA
05F8  1B         1234           DEC DE                ;FINO ALLO ZERO,
05F9  F1         1235           POP AF                ;RIPRISTINA IL CARATTERE
05FA  12         1236           LD (DE),A
05FB  3E3F       1237           LD A,'?'              ;STAMPA UN PUNTO INTERROGATIVO
05FD  CD7707   R 1238           CALL OUTC
0600  97         1239           SUB A
0601  CD8106   R 1240           CALL PRTSTG           ;E IL RESTO DELLA LINEA
0604  C30000   R 1241           JP START
0607  D5         1242  QSORRY   PUSH DE               ;SI PREPARA A STAMPARE 'SORRY'
0608  111E01   R 1243  ASORRY   LD DE,SORRY

                    TINY.BASIC      21$GIUGNO$1979      PAGE  23
 LOC  OBJ CODE M STMT SOURCE STATEMENT                        BAU 4.1

060B  C3DC05   R 1244           JP ERROR
                 1245  ;                                                                                  ;
                 1246  ;                                                                                  ;
                 1247  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 1248  ;                                                                                  ;
                 1249  ;                                                                                  ;
                 1250  ;...................GETLN.................FNDLN.(ET.ALII).........................;
                 1251  ;                                                                                  ;
                 1252  ;                                                                                  ;
                 1253  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                 1254  ;                                                                                  ;
                 1255  ;                                                                                  ;
                 1256  ; 'GETLN' LEGGE UNA LINEA E LA METTE IN 'BUFFER'. APPENA E' CHIAMATA, STAMPA       ;
                 1257  ; IL CARATTERE PRESENTE IN A CHE DEVE ESSERE FORNITO DAL CHIAMANTE. POI            ;
                 1258  ; RIEMPIE IL BUFFER COI CARATTERI PROVENIENTI DA TASTIERA FACENDONE L'ECO.         ;
                 1259  ; IGNORA I LINE-FEED  ED I NULL, FACENDONE PERO' SEMPRE L'ECO.                     ;
                 1260  ; CONTROL-H SERVE A CANCELLARE L'ULTIMO CARATTERE BATTUTO, MENTRE IL RUBOUT        ;
                 1261  ; CANCELLA L'INTERA LINEA E NE ATTENDE UN'ALTRA. IL CR SEGNALA LA FINE DELLA      ;
                 1262  ; LINEA E FA TORNARE 'GETLN' AL PROGRAMMA CHIAMANTE.                               ;
                 1263  ;                                                                                  ;
                 1264  ; 'GETLN' CERCA NELLA SAVEAREA DEL PROGRAMMA D'UTENTE LA LINEA CON NUMERO          ;
                 1265  ; DATO IN HL. DE VIENE USATO COME TEXT POINTER. SE TROVA LA LINEA DE -> ALL'       ;
                 1266  ; INIZIO DI QUESTA (CIOE' AL BYTE CONTENENTE LA PARTE MENO SIGNIFICATIVA DEL       ;
                 1267  ; NUMERO DELLA LINEA), E I FLAG SONO NC E Z. SE LA LINEA CERCATA NON C'E' MA       ;
                 1268  ; E' PRESENTE UNA LINEA CON UN NUMERO MAGGIORE, DE PUNTA A QUEST'ULTIMA ED I       ;
                 1269  ; FLAG SONO POSIZIONATI NC E NZ. INFINE SE ARRIVA AL TERMINE DELLA SAVEAREA        ;
                 1270  ; SENZA AVER TROVATO NE' LA LINEA CERCATA NE' UNA CON NUMERO MAGGIORE,TORNA       ;
                 1271  ; CON I FLAG POSIZIONATI NZ E C.                                                   ;
                 1272  ; 'FNDLN' INIZIALIZZA DE ALL'INIZIO DELLA SAVEAREA DEL PROGRAMMA D'UTENTE PER      ;
                 1273  ; INIZIARE LA RICERCA. CI SONO ALCUNI ENTRY POINT CHE NON INIZIALIZZANO DE.        ;
                 1274  ; 'FNDLNP' COMINCIA LA RICERCA DI UNA LINEA CON NUMERO DATO A PARTIRE DA DE        ;
                 1275  ; 'FNDNXT' INCREMENTA DE DI 2, CERCA UN CR E INIZIA LA RICERCA DELLA LINEA         ;
                 1276  ; CON IL NUMERO DATO.                                                              ;
                 1277  ; 'FNDSKP' CERCA UN CR E POI INIZIA LA RICERCA DELLA LINEA CON NUMERO DATO.        ;
                 1278  ;
                 1279  ;
```

2ª Rassegna
del personal & home computer
e microprocessore
8/11 MAGGIO 1980

U.S. International Marketing Center
(Centro Commerciale Americano)
Via Gattamelata, 5/Milano-Fiera
Orario: 9,30/18
INGRESSO LIBERO

BIT '80 è organizzata dall'U.S. International Marketing Center
e dal Gruppo Editoriale Jackson

Linck

Terra.

Sempre più facile. Un allunaggio è sempre più consueto, in fondo. Ed è sempre più facile per l'uomo disporre di strumenti eccezionali al proprio servizio. Il Personal Computer Apple II fa parte di questi, ed è paragonabile solo a sistemi molto più costosi e ingombranti. Sta su una scrivania, video e stampante compresi. Memoria RAM modulare da 16K espandibile a 64K. Linguaggi BASIC e PASCAL. Collegabile a più floppy disks fino a 1,6 MBytes in linea. 15 colori a bassa risoluzione per grafici o 6 colori ad alta risoluzione. Interfacce per qualsiasi collegamento, anche come terminale intelligente. Ed è facile stupirsi anche del prezzo. Apple II è in vendita, consegna immediata, a L. 1.740.000 IVA compresa. Per avere a portata di mano ogni giorno la soluzione definitiva ai problemi di sempre. Che siano di ricerca, di calcolo, di gestione aziendale. O di count down.*

apple computer

* Apple II è stato scelto dalla NASA per l'operazione spaziale a bordo dello Space Shuttle.

Distribuzione per l'Italia IRET Informatica Via Emilia Santo Stefano 32 Reggio Emilia Tel. 0522.49674 e 41992 Telex 530173 IRETRE